VISION-BASED NAVIGATION AND MAPPING FOR FLIGHT IN GPS-DENIED ENVIRONMENTS

A Thesis
Presented to
The Academic Faculty

by

Allen D. Wu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
December 2010

VISION-BASED NAVIGATION AND MAPPING FOR FLIGHT IN GPS-DENIED ENVIRONMENTS

Approved by:

Professor Eric N. Johnson,
Committee Chair
School of Aerospace Engineering
Georgia Institute of Technology

Professor J.V.R. Prasad
School of Aerospace Engineering
Georgia Institute of Technology

Professor Mark Costello
School of Aerospace Engineering
Georgia Institute of Technology

Doctor Phillip J. Jones
Adaptive Flight Incorporated
Marietta, GA

Professor Eric Feron
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: November 15 2010

To my family.

ACKNOWLEDGEMENTS

I would first like to thank my advisor Dr. Eric Johnson for all his help and guidance through graduate school and for giving me the opportunity to come work at the Georgia Tech UAV Research Facility (UAVRF). During my time here at Georgia Tech, I have learned an immense amount from him. I would also like to thank Dr. Eric Feron for all his help while both at MIT and Georgia Tech. He and Vlad Gavrilets have helped mentor me through the start of my college education and got me interested in working with UAVs in the first place.

The work in this thesis would not have been possible without the contributions of my friends and labmates from the Georgia Tech UAVRF and the Georgia Tech Aerial Robotics team. I would especially like to thank Suresh Kannan, Nimrod Rooz, and Claus Christmann and the research engineers Henrik Christophersen, Jeong Hur, and Wayne Pickell. In alphabetical order: Brendan Andrus, Tony Arkwright, Jeff Baldino, Matt Bigelow, Morten Bisgaard, Hugo De Blauwe, Ainsmar Brown, Girish Chowdhary, Wesley Debusk, Manuh Dhingra, Joel Dunham, Jason Fine, Hal Gates, Stewart Geyer, Jincheol Ha, Hiroyuki Hashimoto, Greg Ivey, Ole Jakobsen, Phillip Jones, Scott Kimbrell, Adrian Koller, Morten Laursen, Keumjin Lee, Paul Mitzlaff, Alex Moodie, Jonathan Muse, Harold Nikoue, Seung Min Oh, John Ottander, Ep Pravitra, Alison Proctor, Adam Rutkowski, Erwan Salaün, Ramachandra Sattigeri, Syed Shah, Mike Sobers, Yoko Watanabe, Adam Wiktor, and Shusaku Yamaura have all been a pleasure to work with during my time here. Michael Kaess from the computer science department at Georgia Techwas also a huge help with the use of vision sensors in this work. I would also like to thank the members from industry that I have had the pleasure of working on projects with including Brent Yates,

Michael Cancienne, Evan Richey, Steen Mogensen, Jim Neidhoefer, Sam Johnson, Andy Shein, and Felipe Bohorquez.

TABLE OF CONTENTS

DEDICATION iii

ACKNOWLEDGEMENTS iv

LIST OF TABLES ix

LIST OF FIGURES x

SUMMARY xiv

I INTRODUCTION 1

1.1 Motivation 2

1.2 Literature Review 4

1.2.1 Summary of Specific Works 7

1.3 Summary of Contributions 10

1.4 Outline of Thesis 11

II BACKGROUND 12

2.1 Relating 3D Position to 2D Camera Images 12

2.1.1 The Basic Pinhole Camera Model 12

2.1.2 Camera Calibration and Lens Distortion 13

2.2 Reference Frames 15

2.3 Overview of the Extended Kalman Filter 16

2.3.1 Extended Kalman Filter Prediction 16

2.3.2 Extended Kalman Filter Correction 16

2.3.3 The Correspondence Problem 18

2.4 Image Processing 19

2.4.1 Detecting Feature Points 19

2.4.2 Stereo Vision 20

III ESTIMATING LANDMARK POSITIONS GIVEN VEHICLE STATES 24

3.1 Extended Kalman Filter Formulation 24

3.1.1 Process Model 24
3.1.2 Measurement Model 25
3.1.3 Initializing Points Located on Ground Plane 26
3.1.4 Initialization of Covariance Matrix for Mapping 27
3.2 Simulation Results for Vision-Based Mapping 28
3.3 Generic Initialization of Database Points 30
IV ESTIMATING VEHICLE STATES GIVEN FEATURE LOCATIONS . 38
4.1 Extended Kalman Filter Formulation 38
4.1.1 Process Model 39
4.1.2 Measurement Model 40
4.2 Application to Indoor Flight 41
4.2.1 Platform and Hardware Description 42
4.2.2 Comparison of Method to Motion Capture System 46
V SIMULATION AND FLIGHT TEST RESULTS 54
5.1 Modifications for Combining the Estimators 55
5.2 Simulation Results 56
5.3 Flight Test Results 60
5.4 Extension to Stereo Vision 62
5.4.1 Simulation Results 64
5.5 Discussion of Method 68
5.5.1 Tuning the Statistical Point Correspondence 68
5.5.2 Implementing the Estimators 68
5.5.3 Camera Calibration 69
5.5.4 Image Processing 70
5.5.5 Computational Complexity 70
VI CONCLUSION 72
6.1 Contributions of Thesis 73
6.1.1 Monocular Vision-Based Mapping 73

6.1.2 Initialization of Points in the Mapping Filter 73
6.1.3 Vision-Aided Inertial Navigation 74
6.1.4 Framework for Flight in GPS-Denied Environments 74
6.2 Future Work 75
APPENDIX A ATTITUDE ERROR STATE REPRESENTATION 77
APPENDIX B DATA NORMALIZATION FOR DLT TRIANGULATION 81
REFERENCES 84

LIST OF TABLES

1 Initialized landmark coordinates in simulation using DLT. 36

2 Effect on disparity in pixels for varying baseline separation distance between cameras and different ranges to target for a stereo rig. The cameras in this have a 320×240 pixel resolution and a 55 degree field of view. 64

LIST OF FIGURES

1 Small unmanned aerial vehicles, such as the Hornet UAV shown above, are often not capable of carrying large computer systems. 3

2 The FCS20 is a small autopilot system with a DSP and FPGA along with integrated sensors including an IMU, GPS, and pressure sensor. 4

3 Camera perspective projection model used for relating 3D position to position in 2D images. The point (A, B, C) is projected onto the camera image plane to the point (b, c). 13

4 An example of the effect of distortion caused by a camera lens. The image on the left is the image obtained directly from a camera with a wide angle lens whereas the right image shows the undistorted version of the image. 14

5 Geometry for a calibrated stereo rig with a baseline distance b between the two cameras with identical focal lengths. A landmark at a distance r from the stereo rig provides a disparity of $D = X_l - X_r$ in the horizontal location between the left and right images. 21

6 The output of the stereo image processor from a synthetically generated scene. Shown in the left image are the locations of the feature points as found in the left camera of the stereo rig. The right image displays the disparity map of the pixels in the left camera to represent the distance to the points. 22

7 Sample rectified images from onboard a rotorcraft UAV using functions from OpenCV. Rectified stereo images simplify the computation of the disparity map for a stereo rig. 23

8 Vectors used in describing the EKF measurement model. 25

9 By assuming all landmarks to lie on the ground, points in the mapping database can be initialized by finding the intersection of the first observation ray with the ground plane. 26

10 Sample images from the simulation of the mapping estimator. Upper left shows the trajectory of the helicopter as it flies around the target points. The upper right window shows the image viewed by the simulated camera. The lower left shows the image processing results of the corner detector. The output of the estimated locations are replicated in the simulated camera image as red spheres in the lower right window. 29

11 Estimate errors for the fifth estimated point. This point is located at [-6, 5.5, 0] ft. 30

12 Progression of the feature estimates in North, East, and down axes for the mapping estimator. The initial estimates are obtained by projecting the initial measurements onto a plane at 20 ft altitude whereas the actual points lie on the ground at 0 ft altitude. 31

13 Visualization for the feature point tracks when initializing the 3D inertial location of landmarks using multiple observations for triangulation. The blue lines show the locations of the last 20 measurements for the associated feature point. 36

14 Visualization for the feature point tracks when initializing the 3D inertial location of landmarks using multiple observations for triangulation. The blue lines show the locations of the last 20 measurements for the associated feature point. 37

15 Ducted fan UAV used for demonstrating the vision-aided inertial navigation algorithm. Tethers are mounted to the side of the aircraft to prevent the aircraft from striking the ground during initial testing. . . 42

16 Avionics block diagram for the ducted fan UAV. Low-level sensor and actuator interfacing and the feedback control are performed on the aircraft's ARM microcontroller, whereas the more computationally intensive image processing and EKF are handled by the GCS laptop. The Vicon motion capture system is for comparison purposes to assess estimator performance. 43

17 Architecture for control of the ducted fan UAV. The outer loop is responsible for tracking position whereas the inner loop regulates attitude of the aircraft. The outer loop commands roll and pitch attitude angles to the inner loop in order to achieve the desired translational motion. 44

18 The experimental setup used for testing the localization and mapping algorithms. The Vicon system consists of the infrared cameras shown in the background. The vision sensor and the IMU are on the black vehicle marked with the reflective markers. A sample target is shown on the ground with black rectangles against a white background. . . . 46

19 Sample output image from the framegrabber and image processing. The detected features are marked by the green crosses. 47

20 Position North for the vehicle when using vision and IMU only for vehicle pose estimation. The Vicon values in this plot have been bias shifted by 0.425 ft. 48

21 Position East for the vehicle when using vision and IMU only for vehicle pose estimation. The Vicon values in this plot have been bias shifted by -0.175 ft. 49

22 Position Down for the vehicle when using vision and IMU only for vehicle pose estimation. The Vicon values in this plot have been bias shifted by 0.25 ft. 49

23 Roll attitude for the vehicle when using vision and IMU only for vehicle pose estimation. 50

24 Pitch attitude for the vehicle when using vision and IMU only for vehicle pose estimation. 50

25 Yaw attitude for the vehicle when using vision and IMU only for vehicle pose estimation. The Vicon values in this plot have been bias shifted by -5.7 degrees. 51

26 Number of points used in estimation. This depends on the number of feature points in the camera's field of view, and which points the state estimator expects to be in the field of view. 51

27 Placing the image processing targets on the ground (configuration A) provided markedly better results than when the target was placed upright on a wall (configuration B). This is because when the camera is looking down at the target, then the gravity vector assists in determining the relative pose of the vehicle. 52

28 Multiple viewpoints from a single camera of a target on the ground. This illustrates the effect of varying light conditions on the feature point detector since the water lying in the ditch causes reflections at specific angles of incidence with respect to the Sun. Several false features are generated by these reflections. 55

29 Demonstration of the removal of points from the database using sample images from a previously recorded flight. The points generated from light reflections off water in a nearby ditch are removed after they have not been seen for several frames. 56

30 The estimated, actual, and commanded positions for the aircraft during a loss-of-GPS scenario expressed in North, East, and altitude coordinates. At $t = 555$ sec, the GPS is ignored from the navigation solution so that only the IMU and monocular vision sensor are being used for determining the location of the helicopter for closed-loop control. . . 58

31 The error between the estimated and actual position of the helicopter during the simulation run. 59

32 The GTMax rotorcraft UAV is a modified Yamaha R-Max helicopter equipped with custom avionics. For testing the vision-based algorithms, a machine vision progressive scan camera is used for acquiring images. Two onboard computers divide up the tasks of image processing and estimation. 60

33 Block diagram illustrating the layout of the dual computers on the GTMax. Image processing is performed on a secondary computer whereas a primary flight computer handles the estimation and control for the aircraft. 60

34 The estimated, actual, and commanded positions for the GTMax during a flight-tested loss-of-GPS scenario expressed in North, East, and altitude coordinates. At $t = 96$ sec, the GPS is ignored from the navigation solution so that only the IMU and monocular vision sensor are being used for determining the location of the helicopter for closed-loop control. 62

35 Simulation setup used for testing the vision-based mapping and navigation filter. A checkerboard pattern is on the face of a building as a sample target. The helicopter UAV hovers in front of the target to map it and then switches over to vision-aided inertial navigation. . . 65

36 A sample output from the stereo image processing taken during a simulation run. The left window shows the locations of the feature points found from the corner detector, and the right window shows a grayscale representation of the computed disparity map. 65

37 The estimated, actual, and commanded positions for the aircraft during a loss-of-GPS scenario with stereo vision expressed in North, East, and altitude coordinates. At $t = 166$ sec, the GPS is ignored from the navigation solution so that only the IMU and stereo vision sensor are being used for determining the pose of the helicopter for closed-loop control 66

38 The error between the estimated and actual position of the helicopter during the simulation run using stereo vision. 67

39 An attitude error quaternion can be used to express the incremental difference between a tracked reference body frame and the actual body frame for the vehicle. 77

SUMMARY

Traditionally, the task of determining aircraft position and attitude for automatic control has been handled by the combination of an inertial measurement unit (IMU) with a Global Positioning System (GPS) receiver. In this configuration, accelerations and angular rates from the IMU can be integrated forward in time, and position updates from the GPS can be used to bound the errors that result from this integration. However, reliance on the reception of GPS signals places artificial constraints on aircraft such as small unmanned aerial vehicles (UAVs) that are otherwise physically capable of operation in indoor, cluttered, or adversarial environments.

Therefore, this work investigates methods for incorporating a monocular vision sensor into a standard avionics suite. Vision sensors possess the potential to extract information about the surrounding environment and determine the locations of features or points of interest. Having mapped out landmarks in an unknown environment, subsequent observations by the vision sensor can in turn be used to resolve aircraft position and orientation while continuing to map out new features.

An extended Kalman filter framework for performing the tasks of vision-based mapping and navigation is presented. Feature points are detected in each image using a Harris corner detector, and these feature measurements are corresponded from frame to frame using a statistical Z-test. When GPS is available, sequential observations of a single landmark point allow the point's location in inertial space to be estimated. When GPS is not available, landmarks that have been sufficiently triangulated can be used for estimating vehicle position and attitude.

Simulation and real-time flight test results for vision-based mapping and navigation are presented to demonstrate feasibility in real-time applications. These methods are then integrated into a practical framework for flight in GPS-denied environments and verified through the autonomous flight of a UAV during a loss-of-GPS scenario. The methodology is also extended to the application of vehicles equipped with stereo vision systems. This framework enables aircraft capable of hovering in place to maintain a bounded pose estimate indefinitely without drift during a GPS outage.

CHAPTER I

INTRODUCTION

This thesis focuses on the application of vision sensors to unmanned aerial vehicles (UAVs) for the tasks of navigation and environment mapping. Navigation or localization is the problem of figuring out where in space a vehicle is actually located and how it is oriented. This information can then be used for the closed-loop control of an aircraft through an autopilot system or for guiding a vehicle from one destination to another. Mapping refers to the problem of determining the relative locations of points or structures of interest in the surrounding environment which can be used for obstacle detection and avoidance or relative navigation. In this work, a method for how a monocular vision sensor can be fused with and inertial measurement unit (IMU) in an extended Kalman filter framework for navigation and mapping is proposed. Here, a Harris corner detector extracts feature points from each captured image, and a statistical Z-test is used to correspond features from frame to frame. Results demonstrating the application to the autonomous flight of a UAV in a loss-of-GPS scenario are presented to validate the method.

The framework presented here consists of a vision-based mapping phase and a vision-aided inertial navigation portion. During nominal operations of a UAV when GPS is available, vision-based mapping can be performed to estimate the locations of landmark points in the surrounding environment. These landmarks are mapped out so that they can in turn be used for estimating the pose of the aircraft should GPS be lost. In the event of GPS outage, vision-aided inertial navigation using observations of the mapped out landmark locations to estimate the position and attitude of the aircraft. For aircraft that are capable of stationary hover, such as rotorcraft, the

framework allows the vehicle to maintain hover indefinitely with a bounded drift in the state estimate when GPS is unavailable.

1.1 Motivation

The combination of an IMU with a Global Positioning System (GPS) receiver has typically been used to determine the position and attitude for an aircraft. In this configuration, accelerations and angular rates from the accelerometers and gyroscopes of the IMU can be integrated forward in time, and position updates from the GPS can be used to bound the errors that result from the integration. This solution to the navigation problem makes aircraft prone to certain modes of failure due to their reliance on the reception of external signals from the GPS network. GPS signals can suffer from obstructions or multipath in cluttered environments, and the reception of these signals can furthermore be jammed or otherwise denied. Similarly, the task of mapping the surrounding environment is commonly approached by using ranging sensors to scan areas of interest. However, these sensors typically rely on the emission and reception of a signal to determine range which can be undesirable if the vehicle needs to remain undetected. Range sensors capable of functioning over longer distances also tend to be large and bulky limiting their applications to smaller aerial vehicles.

Vision sensors have demonstrated immense potential for application to localization and mapping since they provide data about the surrounding environment and simultaneously allow for the possibility of inferring information about vehicle motion from these images. However, the majority of results presented in these areas have been applied to ground robots where size and payload considerations are often not a limitation. Ground robots also have the option to stop motion to safely process information and plan out the next move, whereas aerial vehicles operating in 3D space could crash without pose updates during that time. Over recent years, it has

been proposed that adding an IMU to a vision system could help to assist with these algorithms because the inclusion of inertial sensors allows for the prediction of camera motion from frame to frame and also helps with resolving scale ambiguity. A navigation system that is capable of functioning with only a combination of inertial and vision sensors would be a fully self-contained one that would also not be prone to jamming or detection.

Figure 1: Small unmanned aerial vehicles, such as the Hornet UAV shown above, are often not capable of carrying large computer systems.

There are currently several small UAVs that could benefit from the integration of vision sensors for navigation and mapping purposes. Many of these vehicles in fact already include camera systems since small UAVs are often used for aerial imaging and surveillance. One such example is the Hornet UAV shown in 1 which is capable of close-quarters flying in urban environments. Operating in urban canyons means that this type of vehicle could potentially be prone to GPS dropouts due to obstructions or poor performance from multipath. This small rotorcraft UAV has a miniature autopilot system that was developed at Georgia Tech called the Flight Control System 20 (FCS20) [13] as shown in Figure 2. The FCS20 consists of a processor board

and a sensor board. The processor board contains a signal processor (DSP) and a field-programmable gate array (FPGA). The DSP is a high rate processor that is responsible for performing primary guidance, navigation, and control algorithms. Meanwhile, the FPGA performs the low level hardware interfacing with its numerous digital inputs and outputs and its ability to perform parallel operations. The sensor board has a three axis IMU with accelerometers and gyroscopes, a GPS, and absolute and differential pressure sensors. The FCS20 also has the ability to grab digital images from a high resolution CMOS image sensor. The combination of the camera interface and the FPGA's high rate and parallel processing capabilities make this a potential platform for vision-based algorithms for small-scale UAVs.

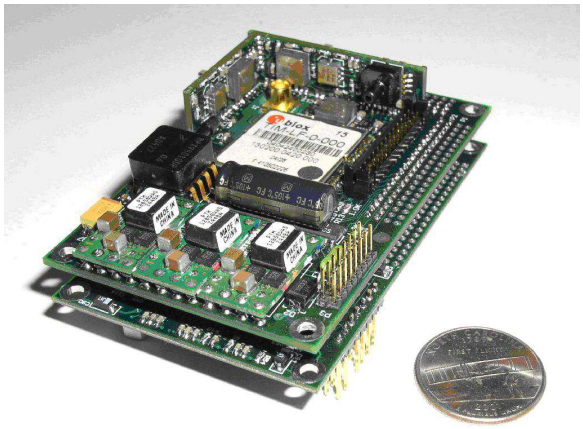

Figure 2: The FCS20 is a small autopilot system with a DSP and FPGA along with integrated sensors including an IMU, GPS, and pressure sensor.

1.2 Literature Review

Many researchers have been investigating the use of vision for localization and mapping onboard UAVs. Some of the initial work in the area started out with the use of stereo vision systems [31]. Stereo systems have seen much success in the areas of vision-based control for ground vehicles, so it seems a natural extension to apply them for use in aerial vehicles [39], [34], [45]. Depth information for a given stereo frame

can be calculated from the disparity between two images. Given the baseline distance between two aligned cameras, the difference in horizontal pixel location of a point in both the left and right images allows the distance to the point to be computed. However, stereo vision systems can be complex due to the amount of image data that needs to be processed as well as the careful camera calibrations and hardware synchronization that needs to be performed. Stereo systems also rely on a minimum baseline distance between the two cameras to provide enough disparity between the two images for adequate ranging information, which may not be practical for many small UAVs.

Some of the earlier results in the area have also included the use of vision to assist in the landing of aerial vehicles such as in [53], [40], [49], and [48]. This is a great use for vision since landing is really a relative navigation problem, especially on mobile platforms such as aircraft carriers or other moving vehicles [5]. Addressing the landing problem also typically means that some knowledge of the environment is available beforehand since the vehicle is usually returning to its original point of deployment. Using the known coordinates or size of targets simplifies the use of monocular vision systems since distance information is implicit in the relative positioning or sizing of features [50], [41]. Researchers have also tried constructing custom targets specifically designed for these applications that allow full pose of the aircraft to be determined from a single observation [60].

Others have tried to take advantage of structured environments for vision-based navigation and mapping with aerial vehicles. If some knowledge of a given environment is available beforehand, then the vehicle can look for more complex shapes than just point features. Most work that used knowledge of the environment's structure has used image processing that would look for lines or squares, both of which are abundant in urban and man-made environments [12]. The use of complex features tends to make the measurements more robust to noise when compared to point

features. However, the image processing also tends to be more computationally intensive, and the requirement that these features be available and abundant enough can be problematic.

An alternative to the use of specific structures in a given environment is to use point features in an environment [46], [14]. Feature points have the advantage that most general environments are capable of providing a rich set of features since point features represent extrema on a very local scale. However, the use of feature points with a monocular vision system for the purposes of navigation and mapping can be difficult for several reasons. First off, observation of a point feature by a single camera only provides the relative bearing to the target. This is because the range information is lost from the projection of the point in 3D space onto the 2D image sensor plane. Furthermore, point features are more sensitive to image noise and distortion because less information is available in each feature point when compared to more complex features. This also makes it more difficult to track or correspond features from frame to frame. These are the challenges that need to be addressed and will be investigated in this work. Researchers have also applied vision sensors to the obstacle detection and avoidance problems [11], [52], [54]. Some have also integrated the trajectory generation of the vehicle together with the visual sensing to optimize the motion of the vehicle for observations from differing viewpoints, such as in [63] and [8], in order to overcome the lack of range information available from monocular vision systems.

Some have investigated vision-based simultaneous localization and mapping (SLAM) for aerial vehicles [32], [10], [64], [56],[25] [44]. In the SLAM problem, both navigation and mapping are performed at the same time so that the vehicle needs to figure out where it is while trying to figure out where the features in space are. The work presented here takes a step back from this problem and instead just looks at the individual problems of localization and mapping separately since there have been limited real-time hardware results in these areas alone. Vision-only algorithms have also been

investigated for the autonomous operation of aerial vehicles such as in [47] and [42].

Optical flow is another vision-based measurement that has been utilized. The optical flow of features from frame to frame is the displacement of a given feature point in the image plane. This represents a velocity type measurement that depends on the relative distance between the camera and the target landmark point. Optical flow has been investigated for aiding with navigation, such as in [59], [30], [55], and [18] as well as for obstacle detection and avoidance [61], [43], [51], [19], [23], and [22].

1.2.1 Summary of Specific Works

There is extensive literature covering the use of vision for mapping and navigation for aerial vehicles currently available. This section summarizes a handful of related works that have already been published to provide a general idea of progress that has been made in the area.

One of the earlier works to appear in the area of vision-based navigation for UAVs was Amidi [4] which presented the development of a visual odometer for flying an autonomous helicopter over flat ground. The visual odometer used a stereo pair of cameras pointed at the ground together with angular rate measurements from gyroscopes to determine the position of the aircraft. The method involved first finding a template, and then tracking the template from frame to frame. Since a stereo pair was used, the relative location of the template with respect to the aircraft could be found from the range information. Position information was backed out from the template track by first removing the vehicle's rotation by using the gyroscopes, and then using the displacement of the template in each image. Flight results using a constrained testbed were presented in addition to results from an outdoor helicopter.

Koch *et al.* have presented work where a downwards looking camera for the closed-loop flight of a helicopter UAV when GPS is lost in [33] which builds upon work done in [7] with the same flight platform. A Lucas-Kanade feature point tracker was used

to detect and correspond features in each image. The location of the landmarks in the environment were found by assuming them to lie on the ground plane. When GPS is lost, then points that have been located can be used for localizing the aircraft in the absence of GPS. Real-time flight test results demonstrated their work.

In [35], Langelaan used an unscented Kalman filter for simultaneously estimating vehicle states as well as landmark locations in inertial space. A Mahalonobis norm was used as a statistical correspondence for data association from frame-to-frame for each estimated landmark. New landmarks were initialized in the filter by performing a projection onto the ground plane. Simulation results for a UAV navigating through a 2D environment were presented.

In [38], Madison *et al.* presented an EKF design where the vehicle states were being estimated as well as the inertial locations of features in 3D space. Their research presented a method which would allow a vehicle with a traditional GPS-aided inertial navigation system (INS) to fly along and estimate the locations of features, and then in the absence of GPS, use these features for aiding the INS. Features were selected and tracked using a Lucas-Kanade feature tracker. A few different methods for initializing the estimated locations of tracked features were implemented. The authors opted for a method of motion stereo where multiple observations of a new feature are first taken, and then the 3–D location of the point is computed using a least-squares solution to find the intersection of the observation rays. Simulation results for a vehicle that momentarily loses GPS were provided. Authors in [3] later built upon this work to fly a quadrotor vehicle autonomously in real-time using offboard processing. A reactive vision-based obstacle avoidance scheme was also incorporated into the system.

The authors in [58] implemented a Harris feature detector along with a random sample consensus (RANSAC) algorithm for the correspondence of features in an FPGA. In this work, Fowers *et al.* used the FPGA to also read in digital images from a digital CMOS camera. This was used for providing drift corrections to an INS

to stabilize a quad-rotor vehicle for short-term hover in an indoor flight environment. By assuming that the the vehicle remains relatively level, the RANSAC algorithm was used to provide estimates of the translation, yaw rotation, and change in scale. RANSAC is a model fitting algorithm where a collection of points are fitted against a model, and the points are sorted into groups of inliers and outliers. These estimated values were then used to provide drift correction measurements to the integration of the IMU.

Frietsch *et al.* in [17] presented an application of vision to the hovering and landing of a quad-rotor UAV. These authors presented a method for estimating the above ground level altitude without the need for ranging sensors by utilizing the optical flow from a vision sensor in conjunction with a barometric altimeter. The proposed method used a Lucas-Kanade tracker to perform the detection and tracking of features from frame to frame. A RANSAC algorithm was then used to estimate a homography that relates points on the ground plane as viewed from two different perspectives. To figure out the distance of the ground plane from the vehicle, the net optical flow of the scene was combined with readings from a barometric altimeter. Since this work was intended for the applications of hovering and landing, it was assumed that the camera's motion was dominated by rotation and that the scene was planar. Simulation results of the method were provided.

Watanabe presented a method for 3D obstacle reconstruction from a 2D monocular vision sensor in [62]. By using image processing to detect line segments in each frame, an EKF was used to estimate the locations of the lines in 3D space assuming knowledge of the vehicle's position and attitude. Data association of lines from frame to frame was handles using a statistical Z-test correspondence. Simulation results for obstacle detection and terrain mapping were provided.

In [65], Webb *et al.* presented an implicit extended Kalman filter that used the epipolar constraint of features from frame to frame to estimate vehicle states. The

epipolar constraint is the coplanarity of the relative observation vectors between two frames. They assumed that a feature point tracker was performing the detection and tracking of features from frame to frame. Simulation results were presented.

Work in [66] presented flight test results where a rotorcraft UAV was able to maintain a stable hover without GPS by fusing image processing results of a target window together with an IMU and a magnetometer in an EKF framework. It was assumed in this preliminary work that the position, size, and orientation of the window were known beforehand. With this information, measurements of the 2D pixel location of the window's center in the camera image as well as its area in the image could be used as sufficient measurements for the estimation. Real-time results of the helicopter flying in front of a target window without GPS for a sustained period of time were presented. Ivey later built upon this work in [24] to investigate its application to the SLAM problem by estimating the position, size, and orientation of the window in parallel.

1.3 Summary of Contributions

The primary contributions of the thesis are as follows:

- Developed a vision-based mapping algorithm capable of running in real-time onboard a UAV with flight imagery obtained from a monocular vision camera. Feasibility of the method is demonstrated with simulation and flight test data.

- Proposed a method for initializing points with multiple observations from a monocular vision sensor. This is validated using flight test sensor data and imagery that are processed offline.

- Vision-aided inertial navigation when the location of landmarks in the surrounding environment are known. This algorithm is capable of being performed in real-time and is applied to the closed-loop flight of a small UAV in an indoor

environment.

- Integrated the mapping and navigation estimators into a framework for handling autonomous flight in GPS-denied environments. This is applied to a loss-of-GPS scenario both in simulation and flight test with a rotorcraft UAV. This methodology is also extended to aircraft equipped with stereo vision systems.

1.4 Outline of Thesis

This thesis starts out by presenting some preliminary information about how measurements can be obtained from vision sensors and an overview of the filtering method employed in this work. With this background information in hand, the following chapter proceeds to describe the algorithm for vision-based mapping when vehicle pose is known. This also includes a proposed method for handling the initialization of the mapped out points in the filter. The following chapter covers the inverse problem of using a vision sensor to figure out the position and orientation of the vehicle when the location of the landmarks in the surrounding environment are known. These two estimators are then combined into a framework that can be used to support aircraft for flight in GPS-denied environments, and this method is extended to stereo vision systems. Results for a loss-of-GPS scenario are presented as well as a discussion of the performance of the proposed framework. A summary of the work and possible future directions are presented in conclusion.

CHAPTER II

BACKGROUND

This chapter provides some background for the problem at hand. First, a description of how the camera is modeled as a pinhole camera is provided. This is followed by a brief overview of the fundamental equations used in the extended Kalman filter. Some modifications are needed for the extended Kalman filter in this problem though to handle the uncertain number of feature measurements in a given frame and for handling the problem of corresponding each measurement to its actual point source in inertial space. These issues are also discussed in the overview of the extended Kalman filter. Finally, a brief discussion of image processing techniques is provided to address how the feature points are extracted from a given image frame.

2.1 Relating 3D Position to 2D Camera Images

A basic pinhole model is often used to model how a point in 3D space is projected onto a 2D camera image. This model, however, often does not accurately reflect the actual behavior of cameras due to the physical optical distortions of the camera lens. In this section, an overview of the pinhole camera model is first presented, followed by an overview of some corrections for camera calibration and lens distortion.

2.1.1 The Basic Pinhole Camera Model

A perspective projection model of a pinhole camera allows position in a 2D camera image to be inferred from 3D position as shown in Figure 3. The model projects an arbitrary point (A, B, C) to a pixel point (b, c) on the image plane (the camera image) according to $b = f(B/A)$ and $c = f(C/A)$ where f is the focal length of the camera.

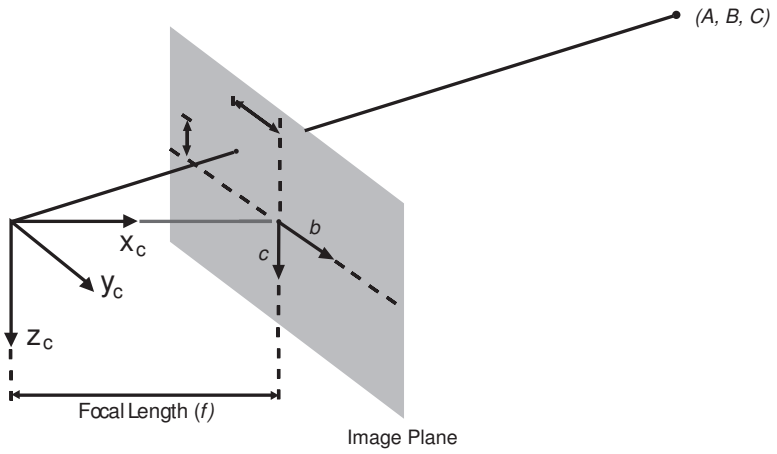

Figure 3: Camera perspective projection model used for relating 3D position to position in 2D images. The point (A, B, C) is projected onto the camera image plane to the point (b, c).

However, in reality, the focal lengths in the horizontal and vertical directions may be different. So these expressions will be rewritten as

$$b = f_x \frac{B}{A} \tag{1}$$

$$c = f_y \frac{C}{A}, \tag{2}$$

where f_x and f_y are the focal lengths in the horizontal and vertical directions respectively. These focal lengths can be computed from knowledge of the width and height of the camera image plane (w and h) and from the horizontal and vertical fields of view (γ_x and γ_y) according to

$$f_x = \frac{w}{2\tan\left(\frac{\gamma_x}{2}\right)}, \qquad f_y = \frac{h}{2\tan\left(\frac{\gamma_y}{2}\right)} \tag{3}$$

2.1.2 Camera Calibration and Lens Distortion

The rectilinear perspective projection model for an ideal camera often does not accurately reflect the behavior of physical cameras due to the effects of lens distortions. Lens optics tend to warp the actual image captured by the camera sensor so that

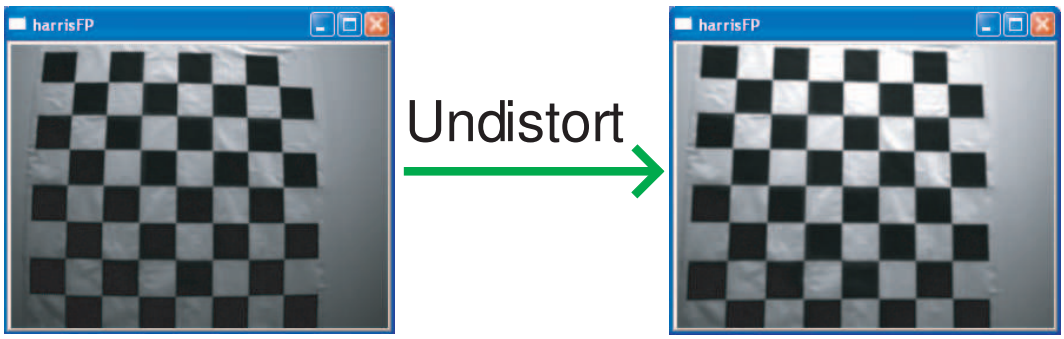

Figure 4: An example of the effect of distortion caused by a camera lens. The image on the left is the image obtained directly from a camera with a wide angle lens whereas the right image shows the undistorted version of the image.

straight lines no longer appear straight in the image. [9] provides a model that is commonly used for representing the undistorted pixel locations after correcting for radial and tangential lens distortions:

$$x_u = x_d\left(1 + k_1 r^2 + k_2 r^4\right) 2p_1 x_d y_d + p_2\left(r^2 + 2{x_d}^2\right) \quad (4)$$

$$y_u = y_d\left(1 + k_1 r^2 + k_2 r^4\right) 2p_2 x_d y_d + p_1\left(r^2 + 2{y_d}^2\right) \quad (5)$$

where $r^2 = {x_d}^2 + {y_d}^2$ and $x_d = X - c_x$, $y_d = Y - c_y$. The k_1 and k_2 parameters are radial distortion coefficients whereas p_1 and p_2 are the tangential distortion coefficients. The offset of the image center is represented by the parameters c_x and c_y in the horizontal and vertical directions respectively. With knowledge of these parameters, the undistorted pixel locations (x_u, y_u) can be computed from location in the raw distorted image obtained directly from the camera (x_d, y_d). The effect of these distortions can be seen in Figure 4 where a sample image of a checkerboard pattern from a wide field of view camera was undistorted using the above correction. In this example, the OpenCV open source computer vision library [1] was used to determine the distortion coefficients and to correct for the lens distortion. The camera calibration routines in OpenCV use different perspectives of a checkerboard pattern to determine the intrinsic parameters for the camera. Parameters provided by the camera calibration are the distortion coefficients k_1, k_2, p_1, and p_2, as well as the focal

lengths f_x and f_y and the offset of the image center as given by c_x and c_y.

2.2 Reference Frames

Three primary frames of reference are needed for this estimation problem. The inertial reference frame is a local inertial frame with its axes aligned in the North, East, and down directions. The camera frame has its origin at the camera's principal point with the x_c axis along the camera's optical axis and the z_c axis pointing downwards. The body frame is fixed to the vehicle center of mass with the x_b axis directed out the nose of the aircraft and the z_b axis pointing downwards. Vector components in the different reference frames can be transformed using direction cosine matrix sequences as follows ([16] and [57]):

$$\mathbf{L}_{cb} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_c & \sin\phi_c \\ 0 & -\sin\phi_c & \cos\phi_c \end{bmatrix} \begin{bmatrix} \cos\theta_c & 0 & -\sin\theta_c \\ 0 & 1 & 0 \\ \sin\theta_c & 0 & \cos\theta_c \end{bmatrix} \begin{bmatrix} \cos\psi_c & \sin\psi_c & 0 \\ -\sin\psi_c & \cos\psi_c & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

$$\mathbf{L}_{bi} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{7}$$

$$\mathbf{L}_{ci} = \mathbf{L}_{cb}\mathbf{L}_{bi} \ . \tag{8}$$

$\mathbf{L}_{cb}$ is a rotation matrix that converts vectors from components in the body frame to components in the camera frame by using the pan (ψ_c), tilt (θ_c), and roll (ϕ_c) angles of the camera. Note, however, that the transformation from the body to the camera frame accounts for only the orientation differences between the two frames. The fact that the camera frame is centered at the camera's location, whereas the body frame is centered at the vehicle center of mass, is neglected. $\mathbf{L}_{bi}$ is a standard rotation matrix from the body to the local inertial frame expressed in quaternions.

The inverse rotations are obtained by swapping the matrix subscript indices and taking the transpose of the appropriate matrix.

2.3 Overview of the Extended Kalman Filter

The EKF formulation used for the vision-based estimation tasks is a mixed continuous-discrete time filter. The EKF algorithm can be broken up into two main phases: prediction and correction. In the prediction phase of the EKF, a nonlinear continuous-time process model is used to propagate the current best state estimate forward in time to come up with a new predicted state estimate. Meanwhile, the correction phase runs at discrete intervals and uses sensors to correct the estimate predicted by the process model. By comparing predicted values of the measurement vector with actual measurements from the image processor, the EKF is able to estimate the desired states.

2.3.1 Extended Kalman Filter Prediction

In the prediction phase of the EKF estimation algorithm, the state estimate $\hat{\mathbf{x}}$ and the covariance matrix $\mathbf{P}$ are updated using a nonlinear model of the vehicle dynamics. The following equations are used for these updates:

$$\dot{\hat{\mathbf{x}}} = f\left(\hat{\mathbf{x}}(t), t\right) \tag{9}$$

$$\dot{\mathbf{P}} = \mathbf{AP} + \mathbf{PA}^T + \mathbf{Q} \tag{10}$$

where $f\left(\hat{\mathbf{x}}(t), t\right)$ is a nonlinear process model for the system dynamics, $\mathbf{A} = \left(\partial f / \partial \hat{\mathbf{x}}\right)|_{\hat{\mathbf{x}}}$ is a Jacobian matrix representing a linearization of the dynamics, and $\mathbf{Q}$ is a positive definite matrix representing the process noise inherent in the system.

2.3.2 Extended Kalman Filter Correction

The EKF makes use of a measurement model $h(\hat{\mathbf{x}}^-)$ that takes in the current best state estimate and computes an expected measurement vector for that given state. By

comparing this expected measurement with the actual measurement from the sensor, corrections for the state estimate and the covariance matrix from the prediction phase of the filter are computed. The equations for these corrections are as follows:

$$\mathbf{K} = \mathbf{P}^{-}\mathbf{C}^{T}(\mathbf{C}\mathbf{P}^{-}\mathbf{C}^{T} + \mathbf{R})^{-1} \tag{11}$$

$$\hat{\mathbf{x}} = \hat{\mathbf{x}}^{-} + \mathbf{K}[\mathbf{z} - h(\hat{\mathbf{x}}^{-})] \tag{12}$$

$$\mathbf{P} = (\mathbf{I} - \mathbf{K}\mathbf{C})\mathbf{P}^{-} \tag{13}$$

where $\mathbf{K}$ is the Kalman gain, $\mathbf{R}$ is a diagonal matrix representing measurement noise in the sensor, and $\mathbf{C} = (\partial\mathbf{z}/\partial\hat{\mathbf{x}})\,|_{\hat{\mathbf{x}}}$ is the Jacobian of the measurement vector with respect to the state vector. Minus superscripts in the above equations denote *a priori* values obtained from the prediction phase equations. The results from (11) - (13) are used by the prediction phase in the next time step to further propagate the state vector and the covariance matrix, and the procedure is repeated.

The EKF corrections can also be performed using a sequential processing of the measurement update. Given the time updated state $\hat{\mathbf{x}}^{-}$ and error covariance matrix $\mathbf{P}^{-}$, and a measurement vector $\mathbf{z}(t_k) = [\mathbf{z}^1(t_k)^T \cdots \mathbf{z}^r(t_k)^T]^T$, it may happen that different components of the measurement vector arrive at different rates or that they may not all be available at a given time step. Applying a sequential measurement update allows each component of the measurement vector to be applied independently of the others as they become available. For $l = 1, 2, \cdots, r$ (r different measurements at time t),

$$\mathbf{K}_k^l = \mathbf{P}_k^{l-1}\mathbf{C}_k^{l\,T}(\hat{\mathbf{x}}_k^{l-1})\left[\mathbf{C}_k^l(\hat{\mathbf{x}}_k^{l-1})\mathbf{P}_k^{l-1}\mathbf{C}_k^{l\,T}(\hat{\mathbf{x}}_k^{l-1}) + \mathbf{R}_k^l\right] \tag{14}$$

$$\hat{\mathbf{x}}_k^l = \hat{\mathbf{x}}_k^{l-1} + \mathbf{K}_k^l\left[\mathbf{z}_k^l - h(\hat{\mathbf{x}}_k^{l-1})\right] \tag{15}$$

$$\mathbf{P}_k^l = \left[\mathbf{I} - \mathbf{K}_k^l\mathbf{C}_k^l\right]\mathbf{P}_k^{l-1} \tag{16}$$

where the starting initial conditions for the sequential measurement update at time $t = t_k$ are $\hat{\mathbf{x}}_k^0 = \hat{\mathbf{x}}_k^-$, $\mathbf{P}_k^0 = \mathbf{P}_k^-$, and the final result of the measurement updates

are $\hat{\mathbf{x}}_k^r = \hat{\mathbf{x}}_k$ and $\mathbf{P}_k^r = \mathbf{P}_k$. For every measurement not available at time $t = t_k$, the measurement update for that step l can be skipped. Whenever a measurement is available at any time instant $t = t_k$, that measurement can be included for this sequential update processing.

2.3.3 The Correspondence Problem

The correspondence problem of relating target measurements to their states or for correlating measurements from frame to frame can be solved using the statistical Z-test. The Z-test uses the state error covariance matrix, $\mathbf{P}$, and the measurement noise matrix, $\mathbf{R}$, to define a Z value that ranks the correlation between the measurement and the target state estimate. The Z value is defined for the EKF as

$$Z = \mathbf{e}^T \left(\mathbf{CPC}^T + \mathbf{R}\right)^{-1} \mathbf{e} \tag{17}$$

where the residual is defined as

$$\mathbf{e} = \mathbf{z} - h(\hat{\mathbf{x}}) \tag{18}$$

so that good matches are indicated by small Z values. Note that the magnitude of Z depends not only on the residual, but also on the covariance matrices, and will be small when $\mathbf{P}$ and $\mathbf{R}$ are large. Therefore, even if the residual is large, great uncertainties in the estimates and the measurements will help to keep the value of Z small. In other words, when the accuracy of the state estimate is poor, the Z-test allows for larger residuals because of the high uncertainty. The Z-test correlates the measurements and estimates by comparing the magnitude of Z to a critical value so that only matches with $Z < Z_{max}$ are considered. Out of the matches that pass the threshold test, the best matching pairs with the lowest Z values are used.

2.4 Image Processing

This section discusses how information can be extracted from a captured image frame. Image processing is an active area of research in its own right since can be challenging to reliably extract large amounts of data in a computationally efficient manner. Here a few feature point based methods that have been widely used in the literature are outlined, and a detailed description of the selected image processing scheme is provided.

2.4.1 Detecting Feature Points

As terminology, the detection of feature points is the determination of feature points in a given frame whereas the tracking of feature points refers to the corresponding of features from frame to frame. Three of the most common methods for detecting feature points are the Kanade-Lucas-Tomasi (KLT) Tracker [28], the Scale-Invariant Feature Transform (SIFT) tracker [37], the Speeded Up Robust Features (SURF) tracker [6], and the Harris corner detector[20]. The KLT, SIFT, and SURF formulations handle both the detection and tracking of feature points. However, these are more computationally intensive than the simple Harris corner detector, and for now, the correspondence (or tracking) of points is handled by the statistical Z-test described above. The Harris corner detector works by first computing a second-order moment matrix of the image intensity gradients (denoted $\mathbf{M}$) which is given by

$$\mathbf{M} = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \tag{19}$$

In other words, for a given pixel, this matrix consists of summations of products of the horizontal and vertical image intensity gradients over a window surrounding the pixel. These summations are taken over 3x3 windows in this work. The following measure is then computed for each pixel

$$M_c = \det(\mathbf{M}) - \kappa \left[\mathrm{Tr}(\mathbf{M})\right]^2 \tag{20}$$

which indicates a feature point if the M_c value is greater than a certain threshold value. This measure looks for strong eigenvalues in more than one direction for corner detection without explicitly computing the eigenvalues, thereby reducing the computational requirements of the image processing. Each image is separated into a uniform grid so that feature points are selected uniformly across each image. A minimum separation distance is also enforced between each selected feature point.

Implementations of a Harris corner detector typically require further selection of the feature points to obtain practical measurements. Since the corner metric is computed over a window of pixels, it is possible for multiple windows around a single corner to all exceed the minimum threshold. In order to get a single reading for each individual corner, a minimum distance is enforced between each point output from the corner detector. Furthermore, it is typically desired that there be a maximum number of corners that are output from the image processor either for memory, processing, or bandwidth limitations. So to ensure that the points output from the corner detector are spread throughout the image, the image can be broken up into a grid with limits on the number of points that can be output for each grid section.

2.4.2 Stereo Vision

Stereo vision systems typically consist of two cameras with a horizontal separation between them known as the baseline distance. For a calibrated stereo rig where the two cameras have coplanar image planes with equal focal lengths and parallel optical axes and undistorted images, this positional offset provides a disparity in the location of objects viewed by the two images as illustrated in Figure 5.

The disparity for a given landmark measurement from the stereo rig is $D = X_l - X_r$ where X_l and X_r are the horizontal pixel locations of the point in the left and right images respectively. Denoting the baseline distance between the two cameras as b,

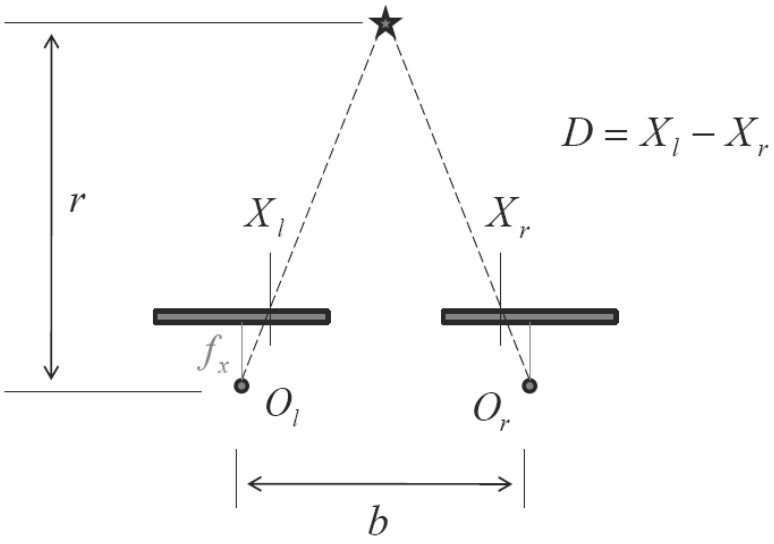

Figure 5: Geometry for a calibrated stereo rig with a baseline distance b between the two cameras with identical focal lengths. A landmark at a distance r from the stereo rig provides a disparity of $D = X_l - X_r$ in the horizontal location between the left and right images.

the relation between range and disparity is given by

$$r = \frac{bf_x}{D} \tag{21}$$

However, most physical stereo rigs do not consist of perfectly aligned identical cameras in reality. Reasons for this can range from imperfect mounting of the individual cameras to manufacturing inconsistencies in the camera and lens assemblies. In order to compensate for these imperfections, a process known as stereo rectification can be employed to generate stereo images that are perfectly calibrated like in the ideal stereo rig in Figure 5. Stereo rectification involves projecting the images from the two cameras onto the exact same plane so that the images are mathematically aligned. In order to perform this rectification, the camera calibration parameters for both cameras need to be known as well as the relative translation and orientation between the two cameras. Therefore, stereo vision systems need to be carefully calibrated in order to easily extract disparity information from them. Rectifying stereo pairs of images also helps to simplify the image processing needed for computing the

disparity for a feature point since the search for the corresponding point need only be performed along the same row in the opposing image.

Figure 6: The output of the stereo image processor from a synthetically generated scene. Shown in the left image are the locations of the feature points as found in the left camera of the stereo rig. The right image displays the disparity map of the pixels in the left camera to represent the distance to the points.

The OpenCV computer vision library was used to perform the stereo calibration and rectification of the stereo image pairs in this work. Bouguet's method for calibrated stereo rectification was used. The also provides a fast block-matching based algorithm for generating a disparity map for all points in the overlap of the stereo image. Stereo image processing in this work consisted of generating the disparity image and finding the feature points in the left camera image. The associated disparities for each of the detected feature points was then looked up from the disparity map to construct a stereo measurement.

Figure 7: Sample rectified images from onboard a rotorcraft UAV using functions from OpenCV. Rectified stereo images simplify the computation of the disparity map for a stereo rig.

CHAPTER III

ESTIMATING LANDMARK POSITIONS GIVEN VEHICLE STATES

This chapter addresses the problem of determining landmark locations when the vehicle pose is known. By mapping out the locations of landmarks in the surrounding environment, this information can either be used for obstacle detection and avoidance or for inferring the pose of the aircraft itself in the event that GPS is no longer available. An extended Kalman filter is used to estimate the 3D coordinates of the landmarks in inertial space. For a monocular vision system, initializing the mapping EKF poses a challenge since only the bearing to the landmark is provided from a single observation and the range information is lost. This chapter discusses the implementation of the EKF for monocular vision systems as well as how the filter can be initialized.

3.1 Extended Kalman Filter Formulation

In this problem, it is assumed that the position ($\mathbf{p}_i$), velocity ($\mathbf{v}_i$), and attitude ($\mathbf{q}$) of the vehicle are known. The states to be estimated are $\mathbf{p}_{fp_i}$ for each feature point to be estimated. The measurements that are used in the EKF are the pixel positions of the feature points in the image plane ($\mathbf{z}_k = [X_k \;\; Y_k]$)

3.1.1 Process Model

Since the feature points are assumed to be stationary, the process model for this problem is very simple. The stationary points have no dynamics, so $f(\hat{\mathbf{x}}) = 0$ and likewise $\mathbf{A} = 0$. This means that the only update that occurs in the prediction phase

for this problem is the update of the covariance matrix according to

$$\dot{\mathbf{P}} = \mathbf{Q} \tag{22}$$

3.1.2 Measurement Model

The measurement model here describes how the expected measurement $\hat{\mathbf{z}} = h(\hat{\mathbf{x}})$ is computed from the propagated state estimate. In order to describe these equations in a succinct manner, the vectors in Figure 8 are first introduced.

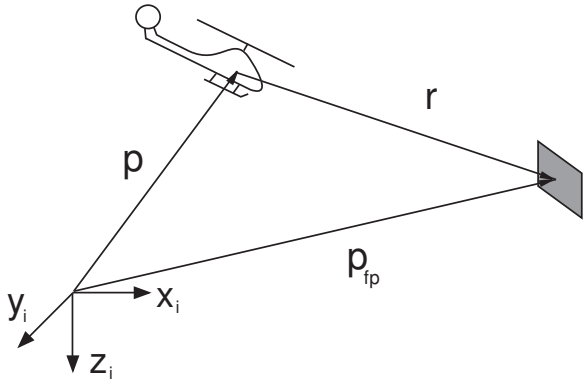

Figure 8: Vectors used in describing the EKF measurement model.

where $\mathbf{p}$ is the position of the vehicle, $\mathbf{p}_{fp}$ is the position of a feature point, and $\mathbf{r}$ is the relative position of the feature point with respect to the vehicle. The relative position vector $\mathbf{r}$ is denoted in the camera frame as $\mathbf{r}_c = [X_{fp_c} \quad Y_{fp_c} \quad Z_{fp_c}]^T$, and similarly in the body frame as $\mathbf{r}_b = [X_{fp_b} \quad Y_{fp_b} \quad Z_{fp_b}]^T$ and in the local inertial frame as $\mathbf{r}_i = [X_{fp_i} \quad Y_{fp_i} \quad Z_{fp_i}]^T$.

For each feature point, the expected measurement is computed as $\hat{\mathbf{z}} = [\hat{X} \quad \hat{Y}]^T$, where from the relations given in (1) and (2) the individual elements are

$$\hat{X} = f_x \frac{\hat{Y}_{fp_c}}{\hat{X}_{fp_c}} \tag{23}$$

$$\hat{Y} = f_y \frac{\hat{Z}_{fp_c}}{\hat{X}_{fp_c}} \tag{24}$$

The following describes the calculations of the partial derivatives needed for computing the Jacobians in the Kalman update from the above measurement model equations.

$$\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{p}}_{fp_i}} = \left(\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{r}}_c}\right)\left(\frac{\partial \hat{\mathbf{r}}_c}{\partial \hat{\mathbf{p}}_{fp_i}}\right) \tag{25}$$

$$\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{r}}_c} = \frac{1}{\hat{X}_{fp_c}}\begin{bmatrix} -\hat{X} & f_x & 0 \\ -\hat{Y} & 0 & f_y \end{bmatrix} \tag{26}$$

$$\frac{\partial \hat{\mathbf{r}}_c}{\partial \hat{\mathbf{p}}_{fp_i}} = \frac{\partial\left(\hat{\mathbf{p}}_{fp_c} - \mathbf{p}_c\right)}{\partial \hat{\mathbf{p}}_{fp_i}} = \frac{\partial \hat{\mathbf{p}}_{fp_c}}{\partial \hat{\mathbf{p}}_{fp_i}} - \frac{\partial \mathbf{p}_c}{\partial \hat{\mathbf{p}}_{fp_i}} = \frac{\partial\left(\mathbf{L}_{ci}\hat{\mathbf{p}}_{fp_i}\right)}{\partial \hat{\mathbf{p}}_{fp_i}} - 0 = \mathbf{L}_{ci} \tag{27}$$

This provides the partial derivatives needed for the **C** matrix in the EKF measurement update equations.

3.1.3 Initializing Points Located on Ground Plane

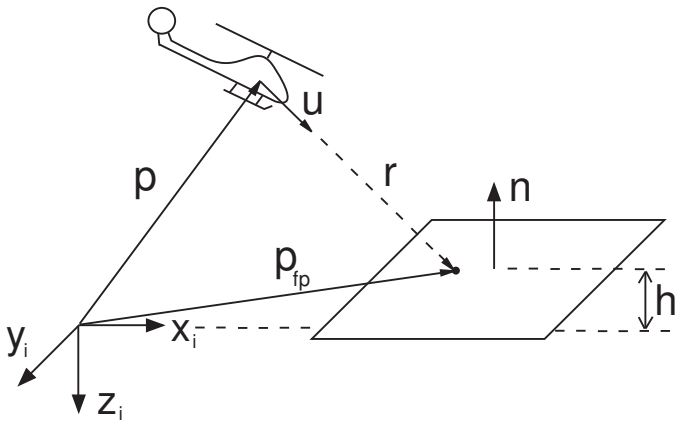

Figure 9: By assuming all landmarks to lie on the ground, points in the mapping database can be initialized by finding the intersection of the first observation ray with the ground plane.

A basic method that can be used to find an initial guess for landmark locations in the EKF is to assume that the points reside on the ground plane. This allows for the

intersection of the ground plane with the ray from the first observation to act as the initial guess when the altitude of the aircraft above the ground can be measured using a rangefinder, or approximated with knowledge of the vehicle's altitude. From the first observation of a point, the vector $\mathbf{u}_i = \mathbf{L}_{ic}[1\ X\ Y]^T$ can be constructed. With this, the relative vector from the vehicle to the feature point can be computed as $\mathbf{r}_i = t\mathbf{u}_i$ where t is the unknown scale factor for the initial observation. To compute, the scale factor, if the ground plane is assumed to be at a height h above the ground, then

$$t = -\frac{(h + \mathbf{p}_{i_z})}{\mathbf{r}_{i_z}} \tag{28}$$

3.1.4 Initialization of Covariance Matrix for Mapping

An initial value for the covariance of the state estimate for each mapped landmark point is required for the EKF algorithm. This can be expressed as

$$\mathbf{P}_{0_c} = Var(\mathbf{r}_{c_x}) \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{R_x}{f_x^2} & 0 \\ 0 & 0 & \frac{R_y}{f_y^2} \end{bmatrix} \tag{29}$$

where $Var\ (\mathbf{r}_{c_x})$ is the variance of the initial range estimate to the landmark point. The variance of the range was modeled as being proportional to the distance to the landmark in an attempt to capture the degradation in range estimation for points located further away. The scaling factor on the range variance term can be tuned as an adjustable parameter. Note however that the initial covariance matrix in (29) is expressed in the camera frame, and since the location of the landmark points are being estimated in the local inertial frame this needs to be transformed into the inertial frame according to

$$\mathbf{P}_{0_l} = \mathbf{L}_{ic}\mathbf{P}_{0_c}\mathbf{L}_{ci} \tag{30}$$

3.2 Simulation Results for Vision-Based Mapping

Preliminary results for the feature mapping estimator were obtained using the Georgia Tech UAV Simulation Tool (GUST) [29] as shown in Figure 10. This simulation environment includes a nonlinear helicopter model in addition to sensor models for the IMU, magnetometer, the vision sensor, and GPS. Synthetic images of the rendered environment are used as image processor inputs. The aircraft model has six rigid-body degrees of freedom and includes dynamic effects from the engine, fuel, landing gear, and rotors. Sensor models include errors, mounting location and orientation, and time delays in the transfer of data. Modeling errors and environmental disturbances, worse than those expected to be encountered during flight, are also injected into the simulation.

Figure 11 shows a sample result for the estimated location of one of the mapped landmark points. In this particular setup, a simulated helicopter flies in a trajectory around two rectangles on the ground to estimate a total of eight landmarks. The trajectory is a square pattern with edges 120 ft in length at 70 ft above ground level and the vehicle is flying with a nominal velocity of 10 ft/s. A simulated 400×300 pixel camera is mounted such that it is angled downwards at a 45 degree angle out the left side of the aircraft. The targets are 12 ft in length and 9 ft in width and are located on the ground with their centers at (0,-10,0) ft and (0,10,0) ft in North-East-down coordinates. The points shown in the plots are the corners of the target centered at (0,10,0) ft. Figure 12 shows the progression of the estimates for both targets in 3-D space. Initialization of the points is performed by projecting the first observations of the points onto the ground plane. In order to observe the convergence properties of the estimator, this simulation example shows the progression of the points when there is an initial error introduced by initializing the points on a plane 20 ft above

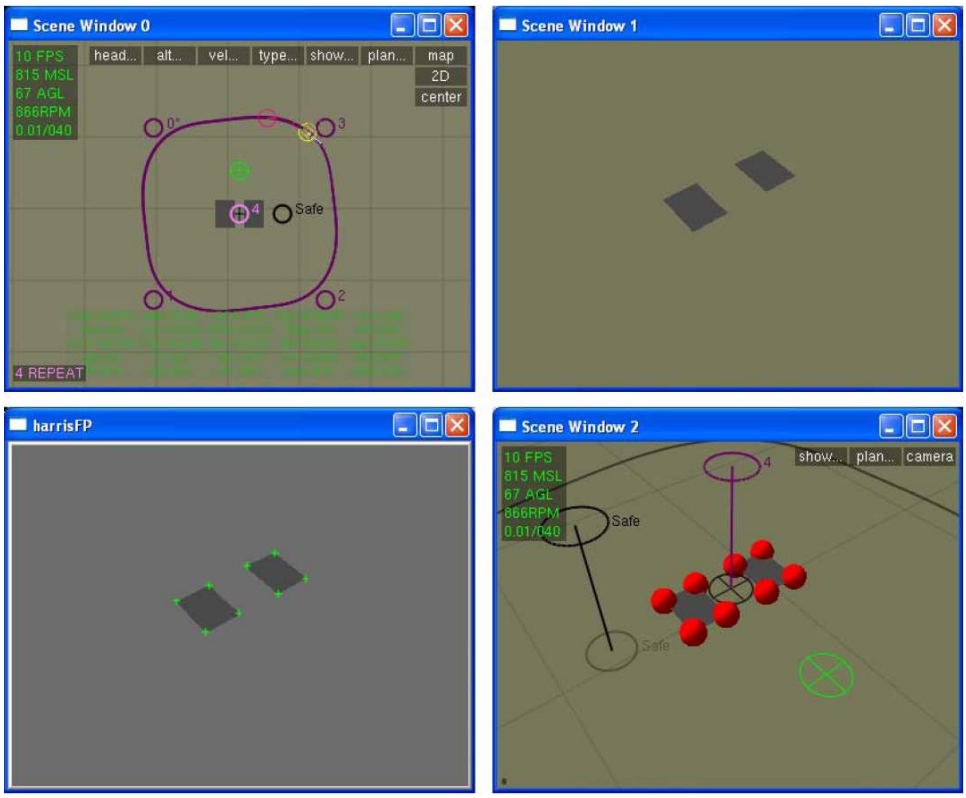

Figure 10: Sample images from the simulation of the mapping estimator. Upper left shows the trajectory of the helicopter as it flies around the target points. The upper right window shows the image viewed by the simulated camera. The lower left shows the image processing results of the corner detector. The output of the estimated locations are replicated in the simulated camera image as red spheres in the lower right window.

the ground.

These simulation results demonstrate the feasibility of the feature point mapping estimator for cases when GPS is available. Convergence of the estimates in this simulation setup is achieved in less than five seconds as can be seen from the plots. This type of mapping performance should be sufficient for obtaining features that could be used for navigating in the event of GPS loss. In these tests, the image processing was executed at only 15 Hz since the simulation was being executed on a single desktop computer so that tasks such as the scene generation occupied additional resources. The computer used for these tests was a Intel Core2 2.4 GHz CPU with 2GB of RAM running Windows XP.

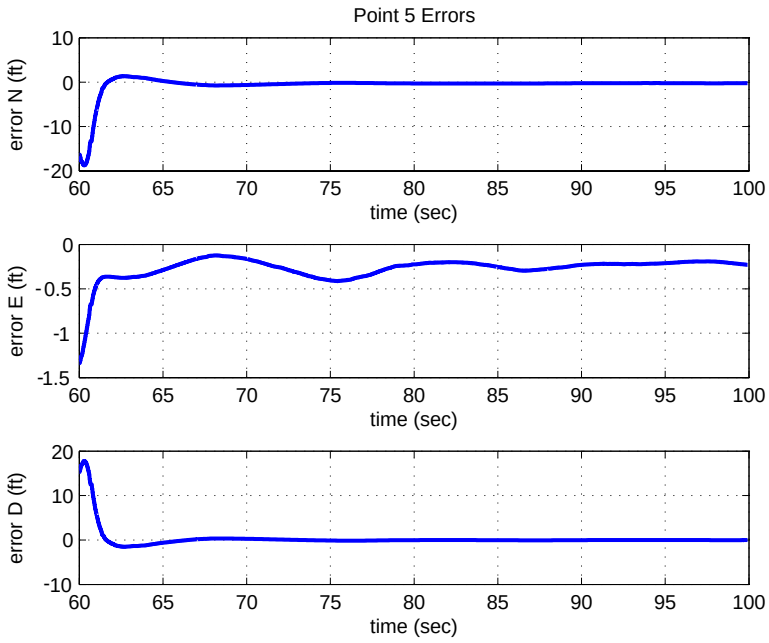

Figure 11: Estimate errors for the fifth estimated point. This point is located at [-6, 5.5, 0] ft.

3.3 Generic Initialization of Database Points

In a monocular vision system, points are projected from 3D space onto a 2D image plane. With this projection comes the loss of range information between the camera and the landmark making it difficult to initialize the states in the mapping filter since the location of a given landmark in inertial space can only be determined along a ray. In the initial work, points were assumed to lie on the ground, so the locations of the points in 3D inertial space were found by intersecting the first observation ray with the ground plane. This method can work well when the landmarks are located close to or on the ground and when the altitude above ground level is known. However it does not work well for initializing points that may lie above the ground such as the walls of buildings.

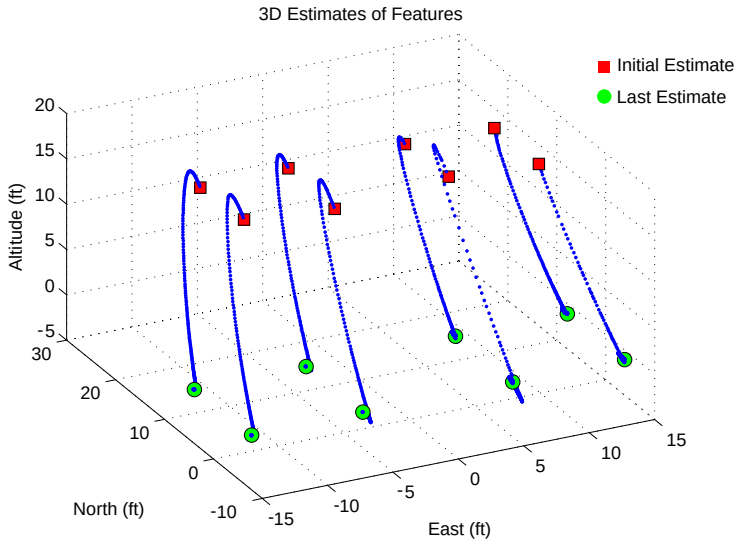

Figure 12: Progression of the feature estimates in North, East, and down axes for the mapping estimator. The initial estimates are obtained by projecting the initial measurements onto a plane at 20 ft altitude whereas the actual points lie on the ground at 0 ft altitude.

To overcome this limitation, multiple observations of a landmark from different perspectives can be used to triangulate its location for the initial estimate. The method employed for this was to correspond the measurements between different images using the knowledge of the change in rotation of the aircraft and to use a template match. Once enough observations of a landmark have been obtained, then a triangulation using the measurements and the vehicle pose at each point in time can be performed.

Measurements can be predicted into the next image using knowledge of the relative attitude change between observations. If the camera calibration matrix is given by

$$\mathbf{\Gamma} = \begin{bmatrix} c_x & f_x & 0 \\ c_y & 0 & f_y \\ 1 & 0 & 0 \end{bmatrix} \tag{31}$$

then the projection of a landmark point onto the image plane can be expressed using homogeneous coordinates. With homogenous coordinates, the coordinate triple

(u, v, w) is equivalent to the 2D point $(u/w, v/w)$ and the coordinate (x, y, z, t) is equivalent to the 3D point $(x/t, y/t, z/t)$. With this representation, the last coordinate essentially acts as a divisor factor that also allows for points to lie at infinity if the divisor factor is 0. Denoting a homogeneous 2D image location of a feature point as $\bar{\mathbf{z}}$ and the homogeneous 3D inertial location of a landmark as $\bar{\mathbf{p}}_{fp_i}$, and by defining the camera matrix $\mathbf{\Phi}$ as

$$\mathbf{\Phi} = \mathbf{\Gamma} \left[\mathbf{L}_{ci} \mid -\mathbf{p}_c \right] \tag{32}$$

the projection of a point onto the image plane can be denoted as

$$\bar{\mathbf{z}} = \mathbf{\Phi} \bar{\mathbf{p}}_{fp_i} \tag{33}$$

An infinite homography can be used to predict the location of a feature point in the next image frame by assuming that the measurement lies on the plane out at infinity. If the first observation is denoted as being in the c_1 frame, and the second observation as being in the c_2 frame, then the infinite homography is such that

$$\bar{\mathbf{z}}_{c_2} = \mathbf{H}_\infty \bar{\mathbf{z}}_{c_1} \tag{34}$$

where $\mathbf{z}_{c_1}$ is the measurement in from the c_1 camera, $\mathbf{z}_{c_2}$ is the measurement as seen from the c_2 camera, and the homography matrix $\mathbf{H}_\infty$ is given by

$$\mathbf{H}_\infty = \mathbf{\Gamma} \mathbf{L}_{c_2 c_1} \mathbf{\Gamma}^{-1} \tag{35}$$

The matrix $\mathbf{L}_{c_2 c_1}$ is a relative rotation matrix from the orientation of the c_1 camera frame to the orientation of the c_2 camera frame so $\mathbf{L}_{c_2 c_1} = \mathbf{L}_{c_2 i} \mathbf{L}_{i c_1}$. This relation allows measurements to be predicted in the next image using only the relative rotation of the vehicle between the two camera poses, which could potentially be provided by gyroscopes, a magnetometer, or other attitude sensing device if a full state estimator is not available.

The direct linear transformation (DLT) algorithm can be used for triangulating the location of a feature point in 3D space from multiple observations using a linear least-squares minimization. This is formulated by starting out with equation (33) which

states that the image projection of a landmark point can be modeled as $\bar{\mathbf{z}} = \mathbf{\Phi}\bar{\mathbf{p}}_{fp_i}$. The homogeneous scale factor can first be removed from this equation using a cross product. For a single image measurement, this yields

$$\begin{bmatrix} 0 & -1 & y \\ 1 & 0 & -x \\ -y & x & 0 \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}^1\bar{\mathbf{p}}_{fp_i} \\ \mathbf{\Phi}^2\bar{\mathbf{p}}_{fp_i} \\ \mathbf{\Phi}^3\bar{\mathbf{p}}_{fp_i} \end{bmatrix} \tag{36}$$

where $\mathbf{\Phi}^i$ is the ith row of the camera matrix $\mathbf{\Phi}$. This can be rewritten as

$$x(\mathbf{\Phi}^3\bar{\mathbf{p}}_{fp_i}) - (\mathbf{\Phi}^1\bar{\mathbf{p}}_{fp_i}) = 0 \tag{37}$$

$$y(\mathbf{\Phi}^3\bar{\mathbf{p}}_{fp_i}) - (\mathbf{\Phi}^2\bar{\mathbf{p}}_{fp_i}) = 0 \tag{38}$$

$$x(\mathbf{\Phi}^2\bar{\mathbf{p}}_{fp_i}) - y(\mathbf{\Phi}^1\bar{\mathbf{p}}_{fp_i}) = 0 \tag{39}$$

However, the third equation can be dropped since it is just a linear combination of the first two. The equations for n different observations of the same landmark are then written in the form $\mathbf{A}_{dlt}\bar{\mathbf{p}}_{fp_i} = \mathbf{0}$, where $\mathbf{A}_{dlt} \in \mathbb{R}^{2n\times 4}$ is given by

$$\mathbf{A}_{dlt} = \begin{bmatrix} x_1\mathbf{\Phi}_1^3 - \mathbf{\Phi}_1^1 \\ y_1\mathbf{\Phi}_1^3 - \mathbf{\Phi}_1^2 \\ \vdots \\ x_n\mathbf{\Phi}_n^3 - \mathbf{\Phi}_n^1 \\ y_n\mathbf{\Phi}_n^3 - \mathbf{\Phi}_n^2 \end{bmatrix} \tag{40}$$

This can be solved by finding the $\bar{\mathbf{p}}_{fp_i}$ that minimizes $||\mathbf{A}_{dlt}\bar{\mathbf{p}}_{fp_i}||$ subject to $||\bar{\mathbf{p}}_{fp_i}|| = 1$. By the Singular Value Decomposition (SVD), it can be written that $\mathbf{A}_{dlt} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ where $\mathbf{U} \in \mathbb{R}^{2n\times 2n}$ and $\mathbf{V} \in \mathbb{R}^{4\times 4}$ are orthogonal matrices, and $\mathbf{D} \in \mathbb{R}^{2n\times 4}$ is a diagonal matrix that contains the singular values of $\mathbf{A}_{dlt}$ in decreasing order. The goal is to minimize $||\mathbf{U}\mathbf{D}\mathbf{V}^T\bar{\mathbf{p}}_{fp_i}|| = ||\mathbf{D}\mathbf{V}^T\bar{\mathbf{p}}_{fp_i}||$ since $\mathbf{U}$ is orthogonal. Furthermore, it is also known that $||\bar{\mathbf{p}}_{fp_i}|| = ||\mathbf{V}^T\bar{\mathbf{p}}_{fp_i}||$ since $\mathbf{V}$ is orthogonal. This means that the problem now becomes to minimize $||\mathbf{D}\mathbf{V}^T\bar{\mathbf{p}}_{fp_i}||$ subject

to $||\mathbf{V}^T\bar{\mathbf{p}}_{fp_i}|| = 1$. A variable substitution can be made by letting $\mathbf{y} = \mathbf{V}^T\bar{\mathbf{p}}_{fp_i}$. This means that the problem can now be formulated as minimizing $||\mathbf{Dy}||$ such that $||\mathbf{y}|| = 1$. Since the singular values in the diagonal $\mathbf{D}$ matrix are sorted in decreasing order, the solution to this problem is for $\mathbf{y}$ to have its last component as 1, and all other components as 0. Solving for $\bar{\mathbf{p}}_{fp_i}$ gives that $\bar{\mathbf{p}}_{fp_i} = \mathbf{Vy} = \mathbf{V}[0\ 0\ 0\ 1]^T$ which means that the minimization problem is solved by having $\bar{\mathbf{p}}_{fp_i}$ equal to the last column of $\mathbf{V}$. Therefore, by solving the SVD of the $\mathbf{A}_{dlt}$ matrix, the last column of the $\mathbf{V}$ matrix provides the triangulated homogenous coordinates of the landmark in inertial space. The data used in the DLT algorithm needs to be normalized as described in Appendix B in order to improve the conditioning of the $\mathbf{A}_{dlt}$ matrix.

The algorithm for the initialization works as follows:

1. Collect all the measurements that were not corresponded from the statistical Z-test method.

2. For all points currently stored in the initialization database, predict these points into the current frame using the infinite homography relation (35).

3. For each of the unmatched measurements, compare it to the list of predicted points from the last image, and find the N nearest neighbors within a box around each measurement.

4. Take the match and perform a template match with the point in the initialization database. The template match score is the sum-square error of the image intensities from a square window around the feature point. The minimum score signifies a best match.

5. Perform this and repeat for all unmatched measurements until each measurement has been corresponded with the best match.

6. Points that found a match can be used to update the initialization database,

whereas measurements that were not matched can be added as new entries into the initialization database.

7. When an entry in the database has M measurements associated with it, then apply the DLT algorithm to the feature track for this landmark point. Compute the sample covariance of the measurements in the feature track with respect to the measurement expected from the computed point (using knowledge of the vehicle pose associated with each measurement in the feature track) according to

$$R_{init} = \sum_{j=1}^{M} \frac{(x_j - x_{j_{exp}})^2 + (y_j - y_{j_{exp}})^2}{M} \tag{41}$$

and add the computed point to the mapping database if $R_{init} <= R_{max}$. Otherwise the point is rejected and the slot in the initialization database can be freed for another point to be tracked for initialization.

8. If a point in the initialization database has not had a new measurement associated with it within the last F frames, then remove the point from the initialization database.

Figure 13 shows a simulation run that was performed to validate the performance of the DLT initialization algorithm. The blue lines show the progression of the feature track over time through the image frame. Targets on the ground have a length of 12 ft and a width of 9 ft with their centers located at $[0, -10, 0]$ ft and $[0, 10, 0]$ ft in North, East, down coordinates. This was performed using the same simulation setup and flight path as was used to verify the mapping estimator. After being initialized, the estimates of the point continued to converge as before using the mapping estimator.

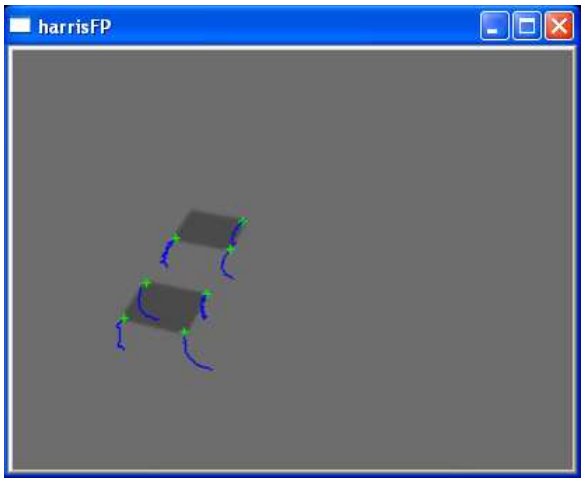

Figure 13: Visualization for the feature point tracks when initializing the 3D inertial location of landmarks using multiple observations for triangulation. The blue lines show the locations of the last 20 measurements for the associated feature point.

Table 1: Initialized landmark coordinates in simulation using DLT.

[ft]	N	E	D	N	E	D
Actual	**6**	**-14.5**	**0**	**6**	**-5.5**	**0**
Estimated	6.77	-13.15	0.45	7.12	-6.20	-1.09
Actual	**6**	**5.5**	**0**	**6**	**14.5**	**0**
Estimated	6.38	8.33	1.95	5.53	17.86	2.70
Actual	**-6**	**-14.5**	**0**	**-6**	**-5.5**	**0**
Estimated	-7.87	-13.54	-1.89	-6.56	-2.28	2.56
Actual	**-6**	**5.5**	**0**	**-6**	**14.5**	**0**
Estimated	-5.94	9.94	4.48	-5.99	18.39	2.51

Using synthetically generated scenes as shown in Figure 13 resulted in frequent mismatches during the nearest neighbor template-matching portion of the algorithm due to the uniformity of the pixel intensities and lack of texture. However, these mismatches did not significantly affect the performance of the initialization method since these points were either rejected by the sample covariance test, or they resulted in an error in the initial guess of the states in the mapping filter which could later be corrected by the estimator. Testing with actual images and sensor data obtained from flight onboard a rotorcraft UAV, as shown in the sample feature tracks in Figure 14, suggested that images obtained from actual cameras are less prone to the occurrence of such mismatches. To date, this initialization method has been tested in simulation

and offline using recorded data from actual flight tests, but not in real-time on actual flight hardware.

Figure 14: Visualization for the feature point tracks when initializing the 3D inertial location of landmarks using multiple observations for triangulation. The blue lines show the locations of the last 20 measurements for the associated feature point.

CHAPTER IV

ESTIMATING VEHICLE STATES GIVEN FEATURE LOCATIONS

This chapter addresses the inverse problem of determining vehicle pose using vision-aided inertial navigation when the landmark positions are already known in inertial space. Being able to localize an aerial vehicle using measurements from a vision sensor allows for the capability to navigate relative to a previously mapped out target and to fly in GPS-denied environments. The coordinates of the landmarks for a given target can either be provided beforehand for a known target, such as a pattern indicating a takeoff or landing area, or the features can alternatively be ones that have already been mapped out by the mapping estimator described in the previous chapter. Additionally, a vehicle intended for flight through a fixed area could have access to a database of positions of recognizable patterns to allow for navigation around a previously marked area. With this *a priori* knowledge of landmarks, a vehicle could navigate autonomously through an area.

4.1 Extended Kalman Filter Formulation

The following are the states that we wish to estimate for the closed-loop control of the aircraft

- vehicle position in inertial space: $\mathbf{p}_i = \begin{bmatrix} p_{x_i} & p_{y_i} & p_{z_i} \end{bmatrix}^T$
- vehicle velocity in inertial space: $\mathbf{v}_i = \begin{bmatrix} v_{x_i} & v_{y_i} & v_{z_i} \end{bmatrix}^T$
- vehicle attitude quaternion: $\mathbf{q} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T$

The measurements we have available from sensor information are:

- body-axis angular rates from the IMU: $\omega_b = \begin{bmatrix} p & q & r \end{bmatrix}^T$
- body-axis specific forces from the IMU: $\mathbf{s}_b = (\mathbf{a}_b - \mathbf{g}_b) = \begin{bmatrix} s_x & s_y & s_z \end{bmatrix}^T$
- pixel position for feature point n: $\mathbf{z}_n = \begin{bmatrix} X_n & Y_n \end{bmatrix}^T$

However, to describe the orientation of the aircraft, an attitude error representation is employed to describe the small angle difference between a reference body frame and the true body-fixed frame. This is accomplished by defining an infinitesimal error quaternion $\delta\mathbf{q}$ according to[36]

$$\delta\mathbf{q} \simeq \begin{bmatrix} 1 & \frac{1}{2}\delta\theta \end{bmatrix}^T \tag{42}$$

such that

$$\delta\mathbf{q} = \mathbf{q} \otimes \hat{\mathbf{q}}^{-1} \tag{43}$$

This allows for $\delta\theta \in \mathbb{R}^3$ to be tracked as a minimal representation of the attitude state error. The accelerometer and gyroscope biases of the IMU are also included in the state vector such that $\omega_b = \omega_{b_{raw}} - \mathbf{b}_g$ and $\mathbf{a}_b = \mathbf{a}_{b_{raw}} - \mathbf{b}_a$. Therefore the estimated 15 element state vector is

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{p}}_i & \hat{\mathbf{v}}_i & \hat{\delta\theta} & \hat{\mathbf{b}}_a & \hat{\mathbf{b}}_g \end{bmatrix}^T$$

where the attitude quaternion $\hat{\mathbf{q}}$ is tracked as the reference body frame to which the attitude error states are compared.

4.1.1 Process Model

The following equations constitute the propagation of the necessary states:

$$\dot{\hat{\mathbf{b}}}_g = \dot{\hat{\mathbf{b}}}_a = 0 \tag{44}$$

$$\dot{\hat{\mathbf{p}}}_i = \hat{\mathbf{v}}_i \tag{45}$$

$$\dot{\hat{\mathbf{v}}}_i = \hat{\mathbf{L}}_{ib}\mathbf{a}_b \tag{46}$$

$$= \hat{\mathbf{L}}_{ib}\left(\mathbf{s}_b + \mathbf{g}_b\right) \tag{47}$$

$$= \hat{\mathbf{L}}_{ib}\mathbf{s}_b + \mathbf{g}_i \tag{48}$$

$$\dot{\hat{\mathbf{q}}} = \frac{1}{2}\begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix}\hat{\mathbf{q}} \tag{49}$$

The following equations are the non-zero components of the Jacobian matrix $\mathbf{A}$:

$$\frac{\partial\dot{\hat{\mathbf{p}}}_i}{\partial\hat{\mathbf{v}}_i} = \mathbf{I}_{3\times3}, \qquad \frac{\partial\dot{\hat{\mathbf{v}}}_i}{\partial(\hat{\delta\theta})} = -\hat{\mathbf{L}}_{ib}\widetilde{\mathbf{s}}_b, \qquad \frac{\partial\dot{\hat{\mathbf{v}}}_i}{\partial\hat{\mathbf{b}_a}} = -\hat{\mathbf{L}}_{ib}, \qquad \frac{\partial(\dot{\hat{\delta\theta}})}{\partial(\hat{\delta\theta})} = -\widetilde{\omega}_b, \qquad \frac{\partial(\dot{\hat{\delta\theta}})}{\partial\hat{\mathbf{b}}_g} = -\mathbf{I}_{3\times3} \tag{50}$$

where the tilde symbol denotes a skew symmetric matrix composed of the vector components such that for the vector $\mathbf{a} = [a_1 \;\; a_2 \;\; a_3]^T$, then $\widetilde{\mathbf{a}}$ is defined by

$$\widetilde{\mathbf{a}} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{51}$$

4.1.2 Measurement Model

The following describes the calculations of the partial derivatives needed for computing the Jacobians in the Kalman update based off of this measurement model. The partial derivatives of the measurement vector with respect to vehicle position in the inertial reference frame is computed as

$$\frac{\partial\hat{\mathbf{z}}}{\partial\hat{\mathbf{p}}_i} = \left(\frac{\partial\hat{\mathbf{z}}}{\partial\hat{\mathbf{r}}_c}\right)\left(\frac{\partial\hat{\mathbf{r}}_c}{\partial\hat{\mathbf{p}}_i}\right) \tag{52}$$

$$\frac{\partial\hat{\mathbf{z}}}{\partial\hat{\mathbf{r}}_c} = \frac{1}{\hat{X}_{fp_c}}\begin{bmatrix} -\hat{X} & f_x & 0 \\ -\hat{Y} & 0 & f_y \end{bmatrix} \tag{53}$$

$$\frac{\partial \hat{\mathbf{r}}_c}{\partial \hat{\mathbf{p}}_i} = \frac{\partial \left(\hat{\mathbf{p}}_{fp_c} - \hat{\mathbf{p}}_c \right)}{\partial \hat{\mathbf{p}}_i} = \frac{\partial \hat{\mathbf{p}}_{fp_c}}{\partial \hat{\mathbf{p}}_i} - \frac{\partial \hat{\mathbf{p}}_c}{\partial \hat{\mathbf{p}}_i} = 0 - \frac{\partial \left(\hat{\mathbf{L}}_{ci} \hat{\mathbf{p}}_i \right)}{\partial \hat{\mathbf{p}}_i} = -\hat{\mathbf{L}}_{ci} \tag{54}$$

The partial derivatives of the measurement vector with respect to the attitude quaternions are similarly computed as

$$\frac{\partial \hat{\mathbf{z}}}{\partial (\hat{\delta\theta})} = \left(\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{r}}_c} \right) \left(\frac{\partial \hat{\mathbf{r}}_c}{\partial (\hat{\delta\theta})} \right) = \left(\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{r}}_c} \right) \frac{\partial \left(\mathbf{L}_{cb} \hat{\mathbf{r}}_b \right)}{\partial (\hat{\delta\theta})} = \left(\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{r}}_c} \right) \mathbf{L}_{cb} \left(\frac{\partial \hat{\mathbf{r}}_b}{\partial (\hat{\delta\theta})} \right) \tag{55}$$

where

$$\frac{\partial \hat{\mathbf{r}}_b}{\partial (\hat{\delta\theta})} = \widetilde{\hat{\mathbf{r}}}_b \tag{56}$$

With these equations for the process and sensor models, it can be shown that at least two landmark points are required for observability of the system in order to estimate vehicle pose and inertial sensor biases.

4.2 Application to Indoor Flight

In order to verify and validate the performance of the vision-aided inertial navigation algorithm, the algorithm was applied to the problem of indoor flight where GPS is not available. For this initial testing, a known target was placed on the ground to generate salient feature points for the vision sensor. Initially, the aircraft was hand-held and manually moved around to simulate motion during flight while running the vision-based estimator, and the outputs from the filter were compared to a high-precision motion capture system. Once performance of the vision-based estimator was verified, then the algorithm was used to hover the vehicle in closed-loop autonomous flight over the visual target. This section provides a description of the hardware setup for the vehicle is provided as well as an overview of the basic controller used to stabilize and maneuver the aircraft. Results from the hand-held testing comparing the vision-based estimator to the motion capture system are also presented.

4.2.1 Platform and Hardware Description

A small 12 inch diameter ducted fan UAV (Figure 15) that weighs about 1 lb was used as a platform for verifying performance of the vision-aided inertial navigation algorithm. This aircraft uses two motors with individual fixed-pitch propellers attached to each one for propulsion and four individually actuated aerodynamic fins as control surfaces suspended below the aircraft in the airflow generated by the propellers. Rigid foam was used for the construction of the duct, and the avionics for the aircraft were mounted on the top right above the duct's intake.

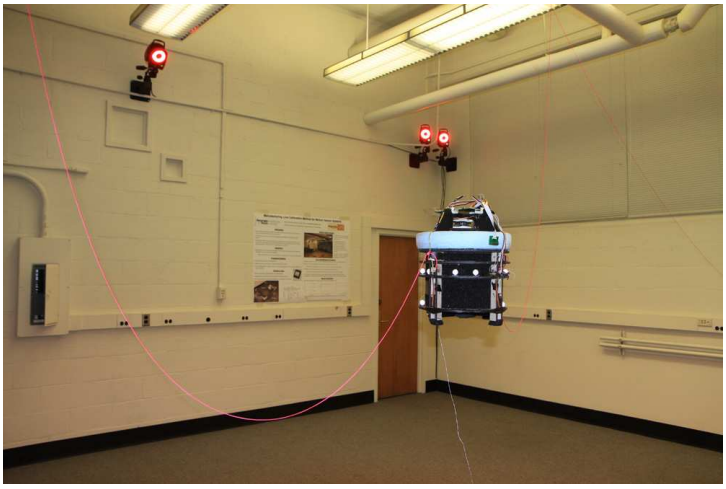

Figure 15: Ducted fan UAV used for demonstrating the vision-aided inertial navigation algorithm. Tethers are mounted to the side of the aircraft to prevent the aircraft from striking the ground during initial testing.

This vehicle uses two counter-rotating propellers for propulsion and aerodynamic vanes for control. However, this UAV is unique in that it is designed to be powered by an offboard supply. The primary advantage of this power architecture is that extra onboard power sources such as fuel or batteries need not be carried by the vehicle. Furthermore, the flight time for this aircraft is limited only by the offboard power supply.

The ducted fan also has a wired video transmission that makes it ideal for image-based algorithms since the frame grabbing and image processing of the video signal can be performed on a ground-based computer. This configuration also helps to keep the vehicle inexpensive and lightweight since the highly capable and expensive hardware reside on the ground. This ducted fan uses a Phillips LPC3180 ARM7 microcontroller for onboard processing and a six degree-of-freedom MicroStrain 3DM-GX1 IMU onboard. A downwards looking NTSC board level CCD camera is mounted beneath the aircraft for the purposes of vision-based navigation. Figure 16 shows how these components are integrated. The ground-based laptop computer uses a Imperx VCE-Pro PCMCIA framegrabber to digitize the analog NTSC video signal.

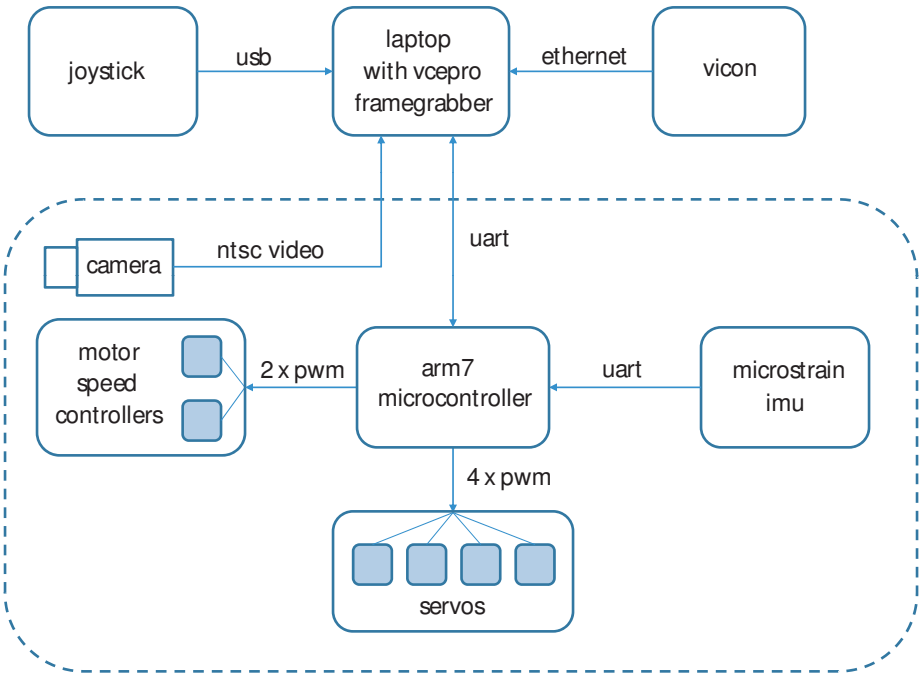

Figure 16: Avionics block diagram for the ducted fan UAV. Low-level sensor and actuator interfacing and the feedback control are performed on the aircraft's ARM microcontroller, whereas the more computationally intensive image processing and EKF are handled by the GCS laptop. The Vicon motion capture system is for comparison purposes to assess estimator performance.

Here, the vision-based estimation algorithm is performed on a ground control station (GCS) laptop. IMU data and the video signal are sent down to the GCS by the vehicle, and the output state information is sent back to the aircraft for use in the vehicle's onboard flight controller. A cascaded outer loop and inner loop architecture was employed for the control architecture. The outer loop was responsible for positional control of the ducted fan aircraft whereas the inner loop controls the attitude of the aircraft. Therefore, in order to achieve a given translational movement, the outer loop commands an attitude to the inner loop. Both the inner and outer loops use a proportional plus integral plus derivative (PID) control scheme. The current control architecture accepts a position in the local inertial reference frame.

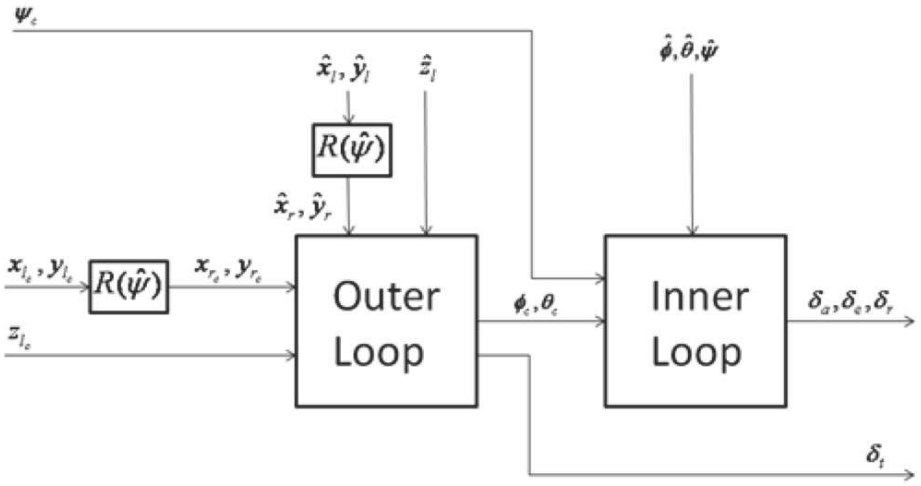

Figure 17: Architecture for control of the ducted fan UAV. The outer loop is responsible for tracking position whereas the inner loop regulates attitude of the aircraft. The outer loop commands roll and pitch attitude angles to the inner loop in order to achieve the desired translational motion.

The symbols in the control architecture diagram are represented as follows:

x_{l_c}, y_{l_c}, z_{l_c}	commanded position in local reference frame
$\hat{x}_l$, $\hat{y}_l$, $\hat{z}_l$	estimated position in local reference frame
x_{r_c}, y_{r_c}	commanded longitudinal and lateral positions after being rotated by heading
$\hat{x}_r$, $\hat{y}_r$	estimated longitudinal and lateral positions after being rotated by heading
ϕ_c, θ_c, ψ_c	commanded roll, pitch, and yaw attitudes
$\hat{\phi}$, $\hat{\theta}$, $\hat{\psi}$	estimated roll, pitch, and yaw attitudes
δ_a, δ_e, δ_r, δ_t	commanded actuator deflections for aileron, elevator, rudder, and throttle
$\mathbf{R}(\hat{\psi})$	2D rotation operation by heading angle to transfrom into body oriented frame

The estimated states in the diagram are output directly from the navigation filter. The equations governing the outer loop control are given by:

$$\delta_t = \delta_{t_0} + K_{P_{alt}} e_{alt} + K_{I_{alt}} \int e_{alt} dt + K_{D_{alt}} \dot{e}_{alt} \tag{57}$$

$$\phi_c = K_{P_{lat}} e_{lat} + K_{I_{lat}} \int e_{lat} dt + K_{D_{lat}} \dot{e}_{lat} \tag{58}$$

$$\theta_c = K_{P_{lon}} e_{lon} + K_{I_{lon}} \int e_{lon} dt + K_{D_{lon}} \dot{e}_{lon} \tag{59}$$

where the errors are defined as $e_{alt} = z_{l_c} = \hat{z}_l$, $e_{lat} = y_{r_c} - \hat{y}_r$, and $e_{lon} = x_{r_c} - \hat{x}_r$. The equations governing the inner loop control are given by

$$\delta_a = K_{P_\phi} e_\phi + K_{I_\phi} \int e_\phi dt + K_{D_\phi} \dot{e}_\phi \tag{60}$$

$$\delta_e = K_{P_\theta} e_\theta + K_{I_\theta} \int e_\theta dt + K_{D_\theta} \dot{e}_\theta \tag{61}$$

$$\delta_r = K_{P_\psi} e_\psi + K_{I_\psi} \int e_\psi dt + K_{D_\psi} \dot{e}_\psi \tag{62}$$

where the errors are defined as $e_\phi = \phi_c - \hat{\phi}$, $e_\theta = \theta_c - \hat{\theta}$, and $e_\psi = \psi_c - \hat{\psi}$.

The aileron and elevator controls are provided by the longitudinal and lateral fins respectively on the ducted fan UAV. Rudder control is realized by a combined motion

of all the fins as well as accelerating one motor and decelerating the complementary motor of the coaxial propulsion assembly.

4.2.2 Comparison of Method to Motion Capture System

Results for the method were obtained using the ducted fan aircraft as shown in Figure 18. This vehicle was marked with reflective markers so that it can be tracked with a high precision Vicon motion capture system. The Vicon motion capture system provides position and attitude measurements of the vehicle at a rate of 100 Hz, and it has been claimed to have on the order of millimeter accuracy. This vehicle was then moved around by hand to simulate motion during these tests.

Figure 18: The experimental setup used for testing the localization and mapping algorithms. The Vicon system consists of the infrared cameras shown in the background. The vision sensor and the IMU are on the black vehicle marked with the reflective markers. A sample target is shown on the ground with black rectangles against a white background.

Targets of black rectangles against a white background were placed on the ground for the camera to look at. The camera and IMU were directly connected to a desktop computer with a dual core 2.4GHz processor with 2GB of RAM. A framegrabber card

digitizes interlaced images from the camera at a rate of 30 frames per second, and the IMU data is read directly over a RS-232 serial port at a rate of 100 Hz. A sample output of the framegrabber and image processing is shown in Figure 19.

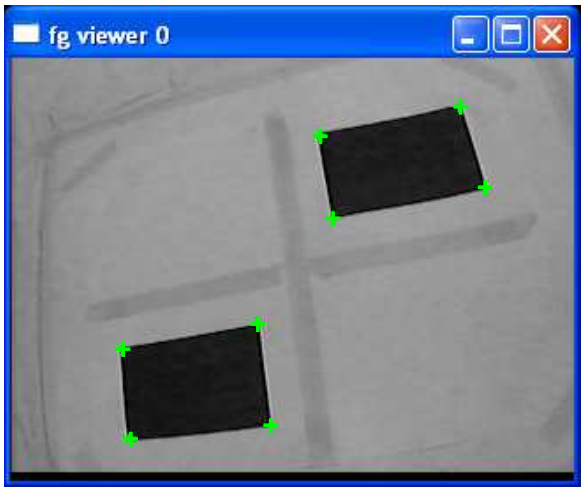

Figure 19: Sample output image from the framegrabber and image processing. The detected features are marked by the green crosses.

In this setup, the targets used are two black rectangles with dimensions 4.5 inches in height and 7 inches in width. The positions of these targets are assumed to be known so that the vehicle can use the locations of the eight total corners for localization. Figures 20 through 26 show results for estimating the vehicle position and attitude using only measurements from the IMU and the monocular camera. Figures 20 through 25 show the estimated positions and attitude angles of the aircraft compared with those provided by the Vicon motion capture system. The results from the Vicon system have been bias shifted by $[0.425 \ -0.175 \ 0.25]^T$ ft in position and -5.7 degrees in heading angle.

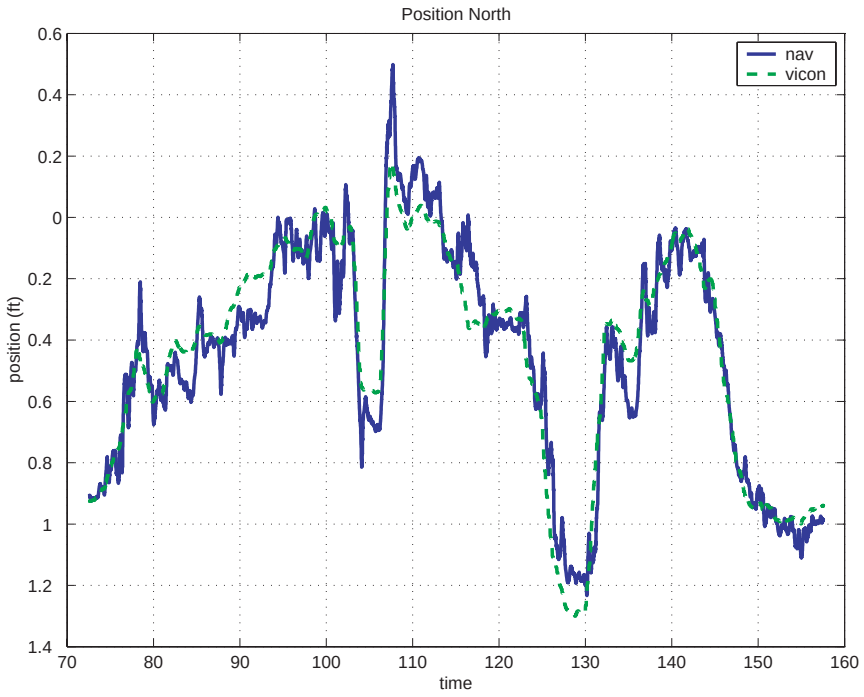

Figure 20: Position North for the vehicle when using vision and IMU only for vehicle pose estimation. The Vicon values in this plot have been bias shifted by 0.425 ft.

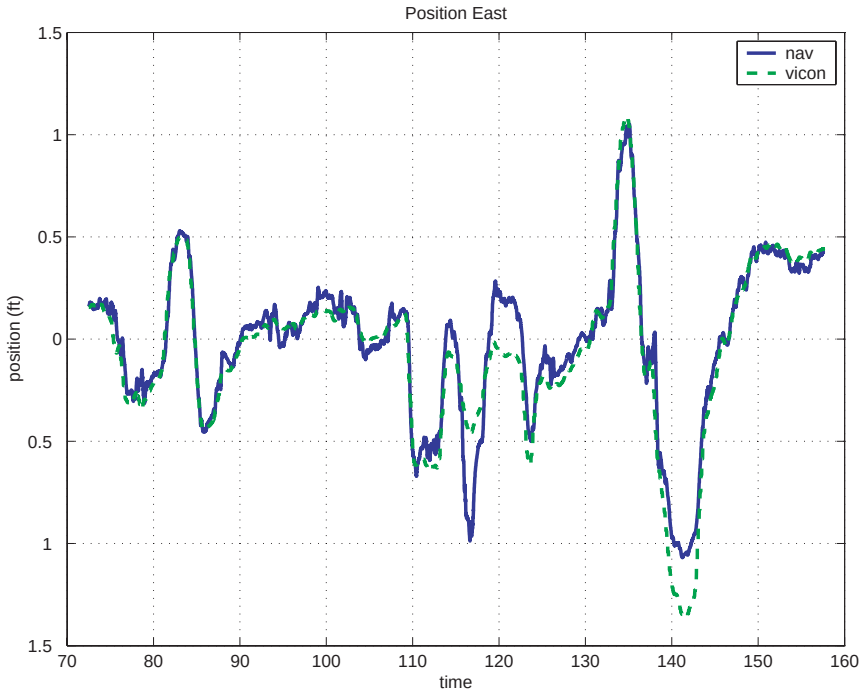

Figure 21: Position East for the vehicle when using vision and IMU only for vehicle pose estimation. The Vicon values in this plot have been bias shifted by -0.175 ft.

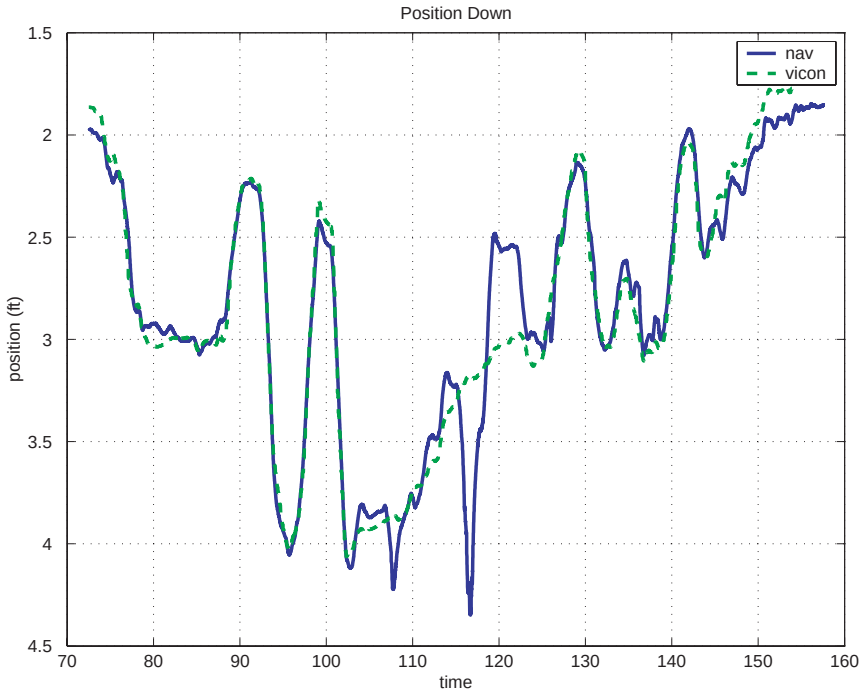

Figure 22: Position Down for the vehicle when using vision and IMU only for vehicle pose estimation. The Vicon values in this plot have been bias shifted by 0.25 ft.

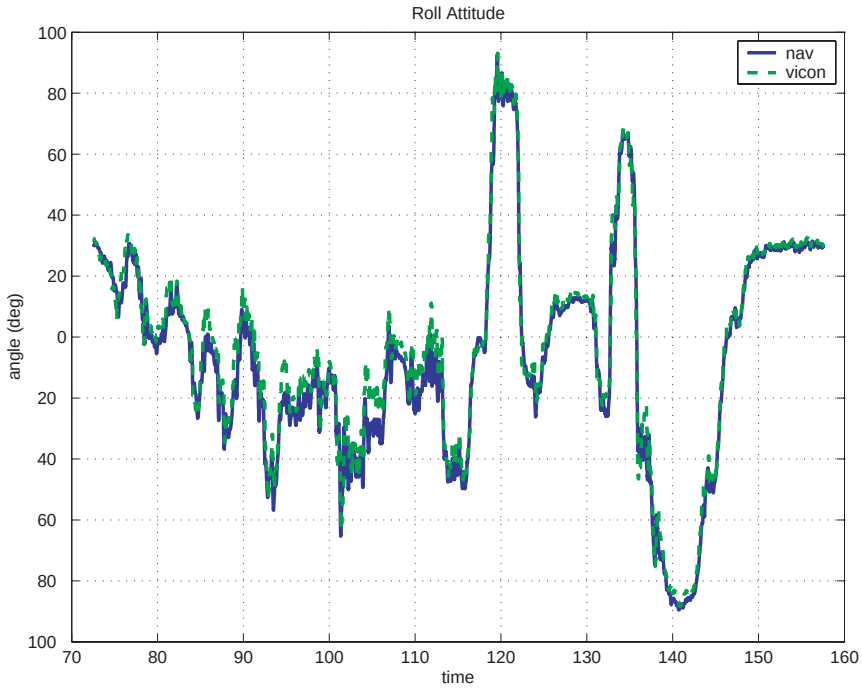

Figure 23: Roll attitude for the vehicle when using vision and IMU only for vehicle pose estimation.

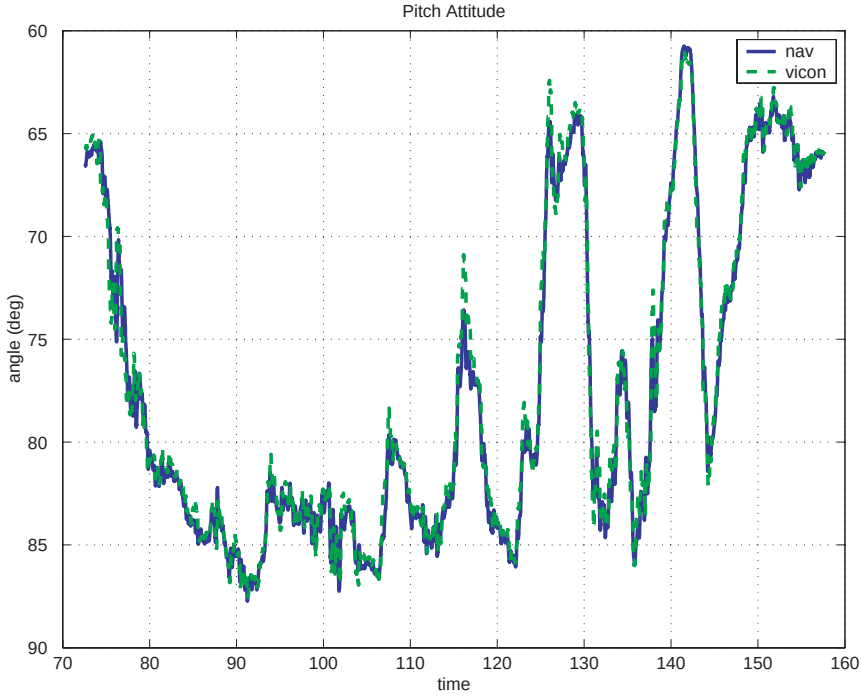

Figure 24: Pitch attitude for the vehicle when using vision and IMU only for vehicle pose estimation.

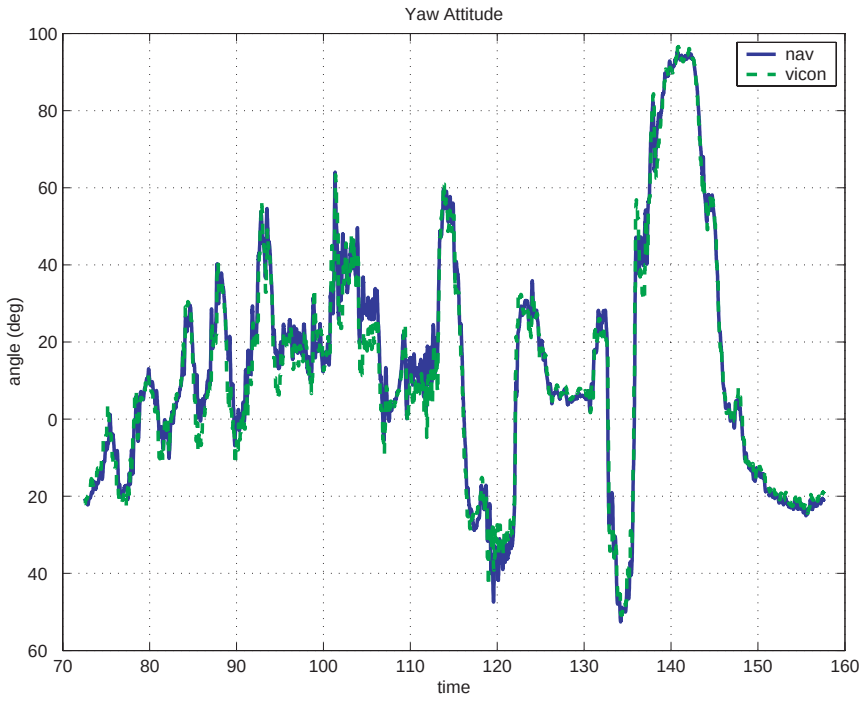

Figure 25: Yaw attitude for the vehicle when using vision and IMU only for vehicle pose estimation. The Vicon values in this plot have been bias shifted by -5.7 degrees.

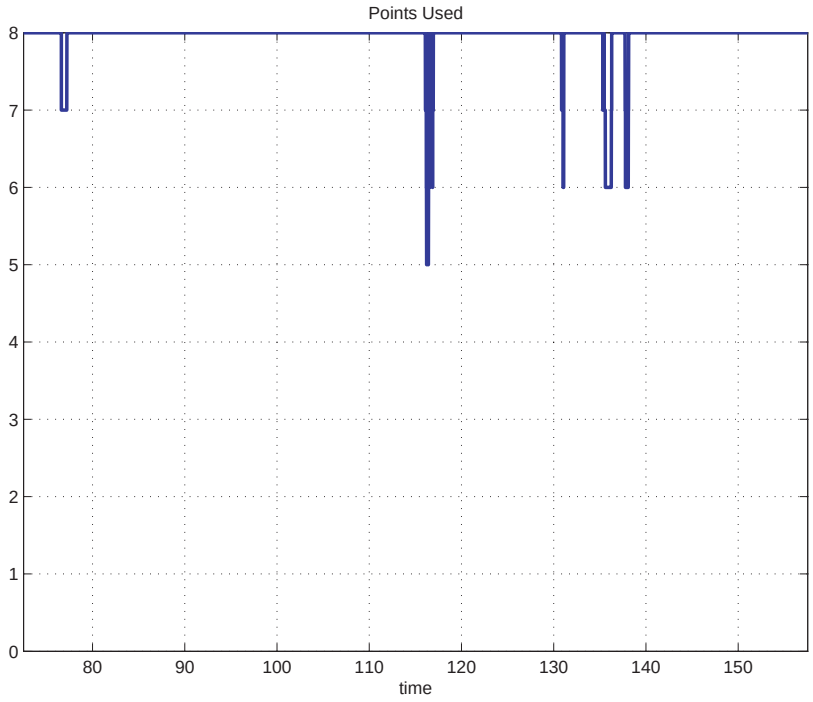

Figure 26: Number of points used in estimation. This depends on the number of feature points in the camera's field of view, and which points the state estimator expects to be in the field of view.

There were a few interesting observations from the hardware implementation of the vision-aided inertial navigation algorithm that are worth noting. It was found that when using these two-dimensional targets, the performance of the localization algorithm was significantly better when the target was placed on the ground as opposed to mounting the target in an upright position such as on a wall (see Figure 27). In fact, it was difficult to get the filter to even converge when the target was mounted on a vertical surface. Conceptually, this occurs because when the vehicle is moving around and looking at the target, the measurements from the image processor may not necessarily change significantly because of the geometry of the problem. Rotation parallel to the target is well determined, but the assistance of the knowledge of the gravity vector is required for the other directions. So when the target is on the ground, gravity helps to resolve this discrepancy by providing information about the pitch and yaw of the vehicle. Similarly, if the target was placed on a vertical surface, a magnetometer could probably assist the filter in a similar manner.

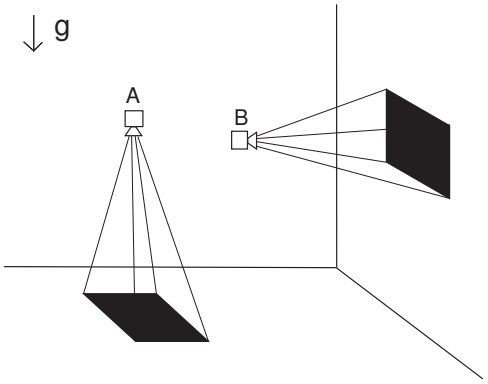

Figure 27: Placing the image processing targets on the ground (configuration A) provided markedly better results than when the target was placed upright on a wall (configuration B). This is because when the camera is looking down at the target, then the gravity vector assists in determining the relative pose of the vehicle.

With the target on the ground, the localization estimator proved to work reliably. Some care did however need to be taken in the initialization of the estimator since

the initial position used by the estimator heavily affected the correspondence of the points. The vehicle needed to be in a position that would correctly associate the measurements with the correct points in the feature database by starting off with each point in the image processor being the closest measurement to the predicted measurement of its associated database point. This should not be a problem for the scenario of GPS loss where usually a good navigation solution is had when GPS is available during nominal operation.

CHAPTER V

SIMULATION AND FLIGHT TEST RESULTS

The mapping and navigation estimators presented in the previous chapters can be combined into a framework that allows for flight in GPS-denied environments. When GPS is available, landmarks can be mapped out using the mapping estimator. Then in the case of a GPS outage, the combined inertial and vision state estimator could be used for stabilizing the vehicle until GPS returns or until the aircraft can maneuver into a position where GPS is available.

A loss-of-GPS scenario was used to demonstrate the proposed architecture in both simulation and flight test. The scenario involved having a rotorcraft UAV flying around a target placed on the ground. While flying around the target, the rotorcraft has knowledge of its position and attitude by means of a baseline GPS-aided inertial navigation system (INS). In this initial stage of the scenario, the mapping estimator is running to figure out the locations of the landmarks in 3D inertial space. When the landmarks have been sufficiently mapped out, the the vehicle is commanded to stop, and the mapping estimator is disabled. With the camera looking at the target, the feature point measurements are then incorporated into the EKF navigation filter. GPS is then ignored in the navigation filter so that the vehicle is performing autonomous closed-loop control using only IMU and the vision sensor. This initial scenario demonstrates the ability of the helicopter to fly around and map out landmarks that it can in turn use to figure out its own position and attitude should GPS be lost. It is also capable of maintaining a bounded drift in the stationary position hold for an indefinite period of time after GPS is lost.

5.1 Modifications for Combining the Estimators

One issue that appeared in the transition from simulation to flight onboard an actual aircraft was that spurious points could appear in the environment momentarily due to noise or variations in ambient lighting. Figure 28 illustrates such a situation during a flight test with different viewpoints from a single camera. In this flight test, the GTMax UAV circles around a target on the ground to obtain multiple perspectives with a downwards looking camera. In this example, a water-filled ditch in the ground generates a reflection at different viewpoints causing spurious points to appear depending on the angle of incidence with respect to the Sun. These points are not desirable to track and may possibly create problems with the correspondence of features from frame-to-frame. In order to remove these points from the database of tracked landmarks, logic was added to remove these points from the database if they have not been associated with measurements after a certain number of frames. Figure 29 demonstrates the performance of this logic when applied to the sample recorded image and data set.

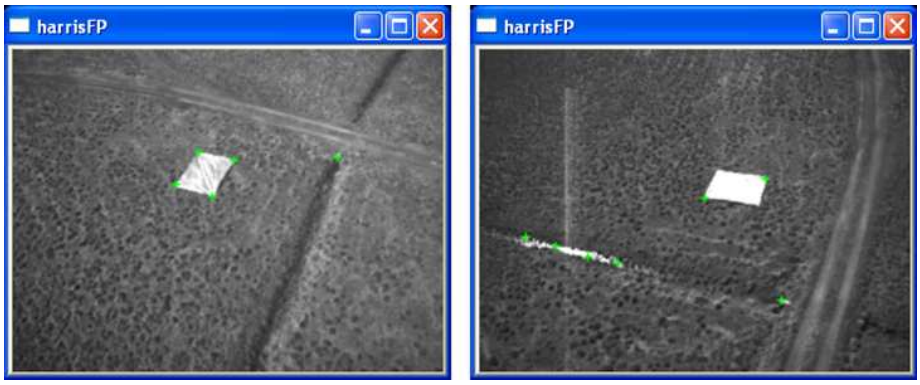

Figure 28: Multiple viewpoints from a single camera of a target on the ground. This illustrates the effect of varying light conditions on the feature point detector since the water lying in the ditch causes reflections at specific angles of incidence with respect to the Sun. Several false features are generated by these reflections.

Figure 29: Demonstration of the removal of points from the database using sample images from a previously recorded flight. The points generated from light reflections off water in a nearby ditch are removed after they have not been seen for several frames.

Additionally, it is desired to use only points that have been sufficiently mapped out by the mapping filter as landmark sources for pose estimation in the navigation filter. The criteria that was used in these tests was a score based on the number of observations corresponded with the point in the mapping database. For every frame that a landmark point has a measurement corresponded with it, the count is incremented. If no measurement has been associated with the database point in a given frame, then the count is decremented. When the estimator is switched over to the navigation mode, then only points that have a count greater than a specified threshold value are actually used to estimate aircraft pose.

5.2 Simulation Results

To initially validate the architecture, simulations were first performed using the GUST simulation tool shown in Figure 10. In the sample simulation scenario, a rotorcraft UAV is circling around some objects of interest located on the ground using a GPS-aided INS. A simulated monocular vision camera with a resolution of 400×300 pixels is looking out the left side of the aircraft and downwards at a 45 degree angle. As it circles these objects, it uses the measurements from the monocular camera as well as the knowledge of its position and attitude as given by the GPS-aided INS to map out

where the landmarks are located in 3D space. Landmarks are initialized by assuming them to lie on the ground plane at 0 ft altitude. The estimates of the landmark locations are represented by the red spheres in the scene windows. After these points have been sufficiently mapped out, the vehicle is placed in a stationary hover, and then GPS readings are ignored so that the vision sensor takes the place of the GPS in the navigation filter. The vehicle is able to maintain a stable closed-loop hover using the vision-aided INS. The helicopter also maneuvers in different directions to demonstrate the transient response of the system. The plot in Figure 30 compares the output from the vision-based estimator with the commanded position as well as the true position as given by the dynamic model of the simulation. It should also be noted that the state output of the vision-based estimator is used in the closed loop control of the aircraft in this simulation test. Figure 31 shows the error between the estimated and actual positions of the helicopter in simulation.

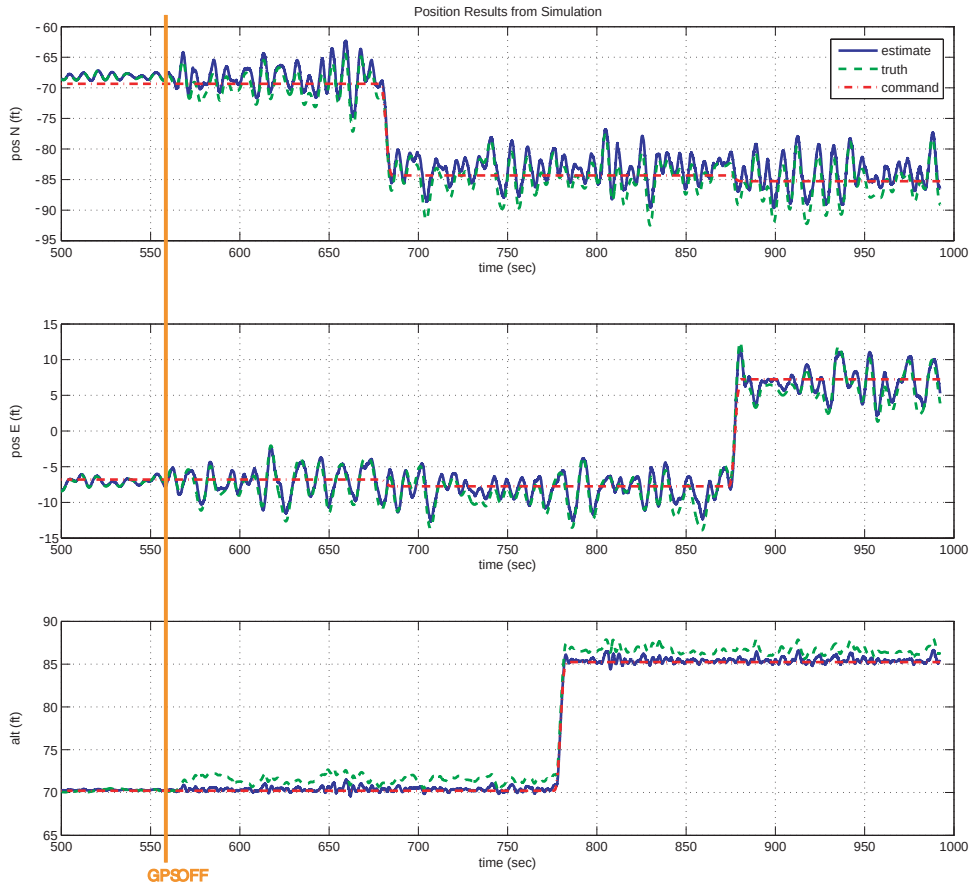

Figure 30: The estimated, actual, and commanded positions for the aircraft during a loss-of-GPS scenario expressed in North, East, and altitude coordinates. At $t = 555$ sec, the GPS is ignored from the navigation solution so that only the IMU and monocular vision sensor are being used for determining the location of the helicopter for closed-loop control.

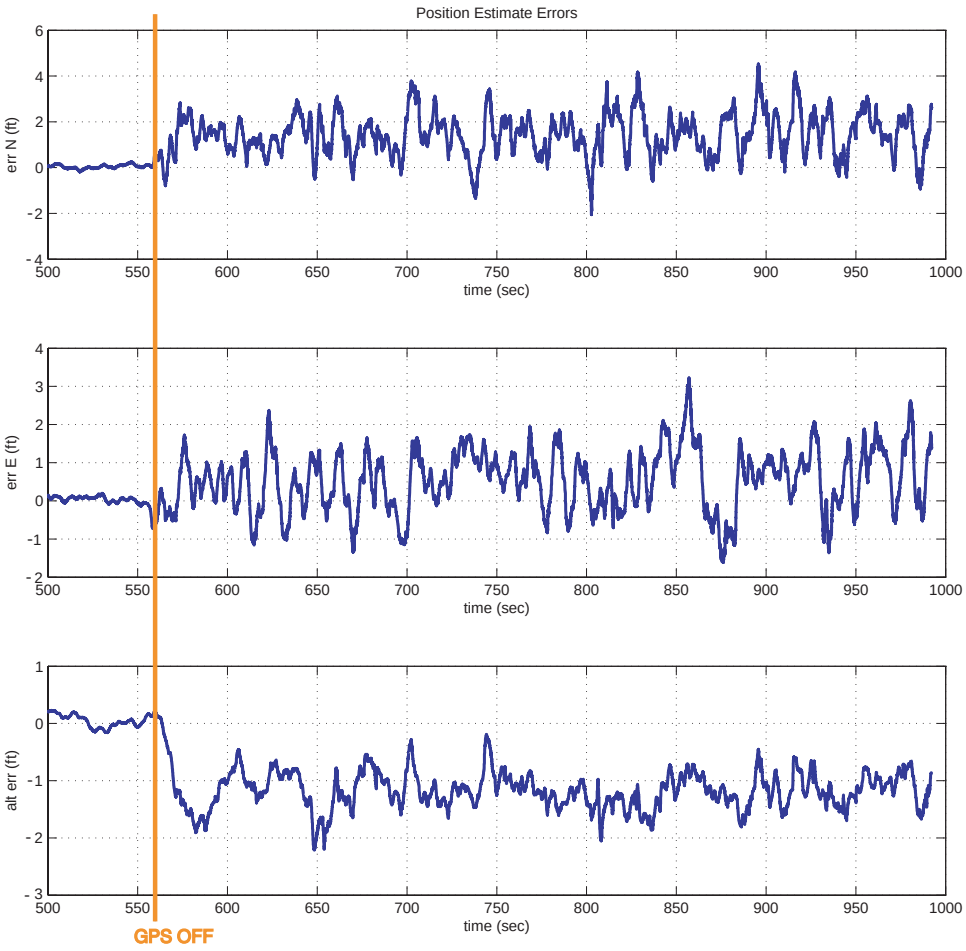

Figure 31: The error between the estimated and actual position of the helicopter during the simulation run.

5.3 Flight Test Results

Figure 32: The GTMax rotorcraft UAV is a modified Yamaha R-Max helicopter equipped with custom avionics. For testing the vision-based algorithms, a machine vision progressive scan camera is used for acquiring images. Two onboard computers divide up the tasks of image processing and estimation.

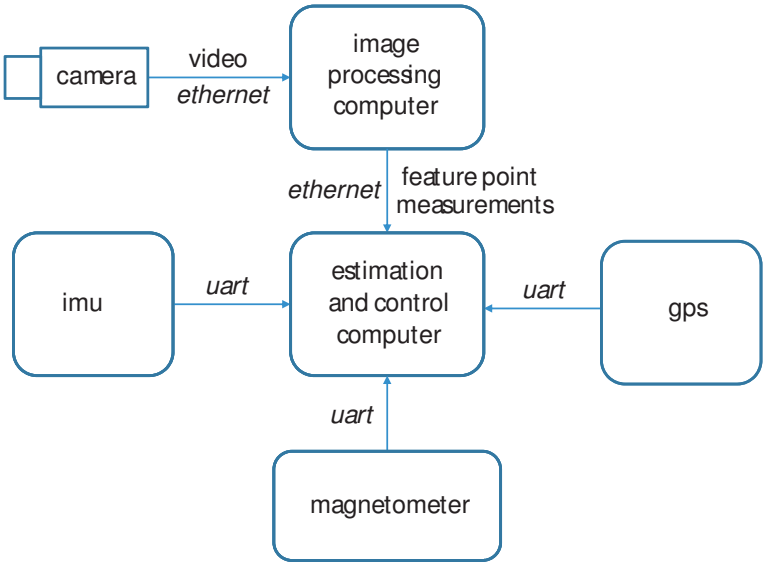

Figure 33: Block diagram illustrating the layout of the dual computers on the GTMax. Image processing is performed on a secondary computer whereas a primary flight computer handles the estimation and control for the aircraft.

After having validated the performance of the system in a loss-of-GPS scenario in simulation, the vision-based mapping and localization algorithms were also verified in

real-time onboard a rotorcraft UAV during a flight test to demonstrate feasibility of the proposed architecture on actual hardware. Georgia Tech's GTMax UAV was used for these tests as shown in Figure 32 [27]. The GTMax is a modified Yamaha R-Max helicopter UAV that is equipped with dual flight computers, an Inertial Science ISIS IMU, a Honeywell HMR2300 magnetometer, and a Novatel OEM4 differential GPS receiver. An adaptive neural-network feedback controller uses EKF state estimates, obtained from the sensor array, to guide the helicopter appropriately [26]. The GTMax is under 200 lbs in weight and has a 6 ft rotor diameter. Dual flight computers divide up the processing for the test by having the primary flight computer perform the estimation and control of the aircraft while the secondary flight computer handles the acquisition and processing of images. For these tests, an ethernet progressive scan camera designed for machine vision is also included as an onboard sensor as shown in Figure 32. The use of a progressive scan camera helps to eliminate any potential problems that may arise from the interlaced images of lower quality video cameras. This camera is mounted on the nose of the aircraft pointing towards the front and angled downwards at a 45 degree angle. The camera provides monochrome images at a resolution of 320×240 pixels and the onboard image processing computer was able to handle this data at a rate of 20 Hz. PC104 embedded computers were used for the dual flight computers. Each one has a Pentium M 1.8 GHz processor and 1GB of RAM running Linux OS.

A similar scenario to the simulation setup was performed with the GTMax in flight. A single 12×9 ft silver rectangle was placed on the ground to act as a target and provide 4 relatively consistent landmarks. The helicopter flies in a circular pattern around the target at an altitude of 90 ft. Once the points have been sufficiently mapped out, then the helicopter is commanded to a stop, and the vision-based estimator is switched from mapping mode to navigation mode. The GPS is then ignored

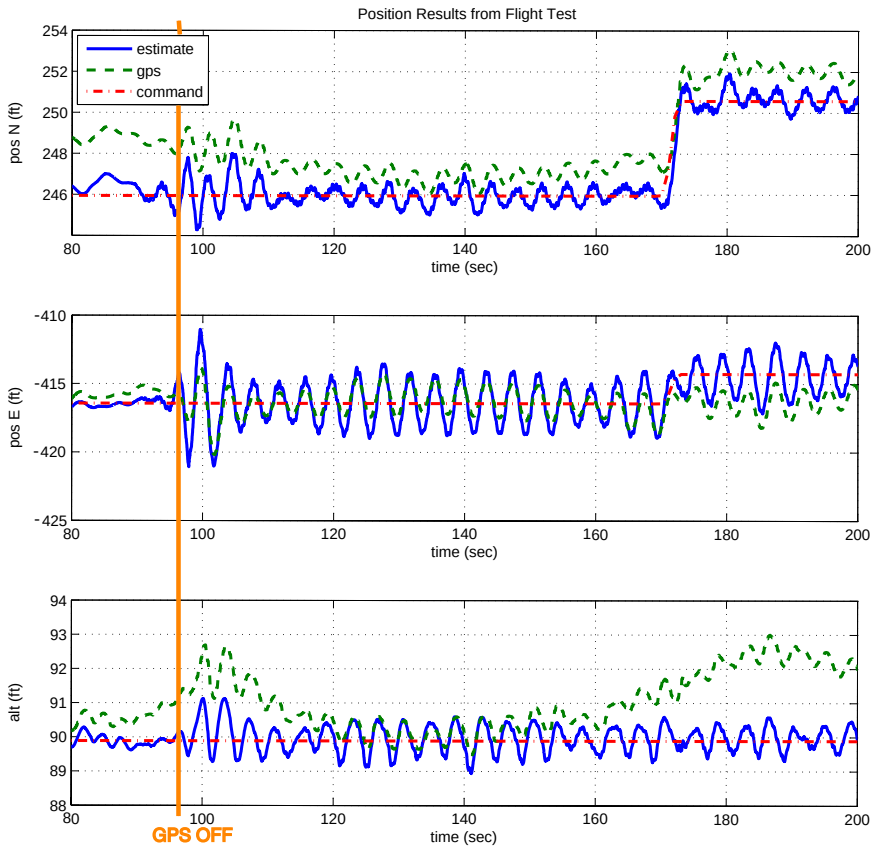

Figure 34: The estimated, actual, and commanded positions for the GTMax during a flight-tested loss-of-GPS scenario expressed in North, East, and altitude coordinates. At $t = 96$ sec, the GPS is ignored from the navigation solution so that only the IMU and monocular vision sensor are being used for determining the location of the helicopter for closed-loop control.

from the EKF so that the helicopter is using only the IMU and vision sensor for determining the states needed for closed-loop control. Figure 34 shows the estimated and commanded positions of the aircraft compared to the position provided by the GPS during this test. A 5 ft translational motion is also commanded to the aircraft while under vision and IMU only control.

5.4 Extension to Stereo Vision

The proposed architecture of using vision-based mapping and navigation for handling loss-of-GPS scenarios can easily be extended to vehicles equipped with a stereo vision system with some minor adjustments. When stereo information is available, then the

expected measurement vector becomes $\hat{\mathbf{z}} = [\hat{X} \quad \hat{Y} \quad \hat{D}]^T$ where from equation (21) the disparity $\hat{D}$ is given by

$$\hat{D} = \frac{bf_x}{\hat{X}_c} \tag{63}$$

and b is the baseline separation distance between the left and right cameras. These stereo measurements can be incorporated into the EKF by using this expanded measurement vector and by modifying the Jacobian of the measurement with respect to the states in equations (25) and (52). The only modification that needs to be made is to the $(\partial\hat{\mathbf{z}}/\partial\hat{\mathbf{r}}_c)$ which now contains an additional row consisting of

$$\frac{\partial\hat{D}}{\partial\hat{\mathbf{r}}_c} = \begin{bmatrix} \frac{-bf_x}{\hat{X}^2_{fp_c}} & 0 & 0 \end{bmatrix} \tag{64}$$

Stereo information also assists with the initialization of points in the mapping database since the disparity information from the stereo rig provides relative range information to the landmark point. This allows for the 3D location of a landmark point to be determined from a single observation. Even with the extra range information from the stereo system, two landmark points are still required for observability of vehicle pose and inertial sensor biases.

Table 2 illustrates the effect of distance and baseline separation between cameras in a stereo rig on the disparity observed by the system. This table uses a sample stereo rig where each camera has a 320×240 pixel resolution and a 55 degree field of view. It can be seen that as the distance increases, the disparity rapidly decreases as well as the amount of change in disparity with respect to changes in distance. This suggests that beyond a certain range, it may not be desirable to use disparity measurements from the stereo rig since it the range estimates will be highly sensitive to noise in the disparity measurement. So in this framework, if the disparity is below a minimum value, then the measurement can be treated as a monocular vision measurement and the nominal monocular vision algorithm and measurement model can be applied.

Table 2: Effect on disparity in pixels for varying baseline separation distance between cameras and different ranges to target for a stereo rig. The cameras in this have a 320×240 pixel resolution and a 55 degree field of view.

	Baseline (ft)				
Range (ft)	**0.5**	**1**	**3**	**5**	**10**
25	6.15	12.29	36.88	61.47	122.94
50	3.07	6.15	18.44	30.74	61.47
75	2.05	4.10	12.29	20.49	40.98
100	1.54	3.07	9.22	15.37	30.74
125	1.23	2.46	7.38	12.29	24.59
150	1.02	2.05	6.15	10.25	20.49
175	0.88	1.76	5.27	8.78	17.56
200	0.77	1.54	4.61	7.68	15.37
225	0.68	1.37	4.10	6.83	13.66
250	0.61	1.23	3.69	6.15	12.29

5.4.1 Simulation Results

To test the application of the mapping and navigation filter for a loss-of-GPS scenario with a stereo vision rig, the simulation setup shown in Figure 35 was used. In this scenario, the stereo rig on the simulated vehicle has a baseline distance of 3.28 ft between the two cameras. Both cameras are angled downwards at a 45 degree angle out the nose of the aircraft. These simulated cameras capture scenes at a 400×300 resolution and are both perfectly calibrated since they are capturing computer generated scenes. A checkerboard pattern target is mounted on the wall of a building to generate clean feature points. The checkerboard pattern consists of 2 rows of black and white squares with edges of 2.5 ft for each side.

The left window in Figure 36 shows the detected feature points and the right window shows a visualization of the computed disparity map. Lighter portions in the grayscale disparity map represent points that are closer whereas the blacked out regions are places where a valid disparity could not be computed.

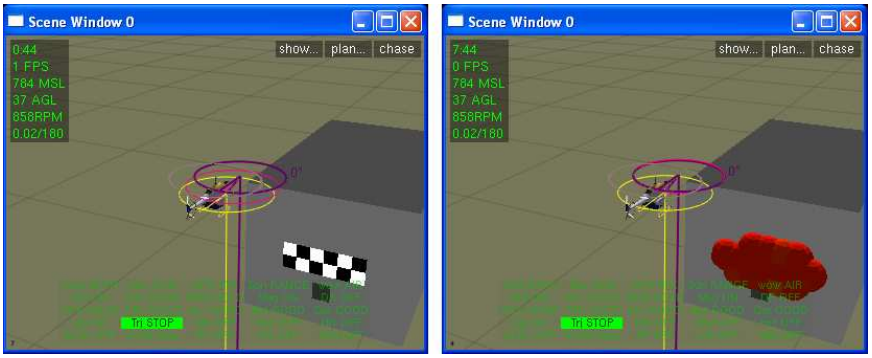

Figure 35: Simulation setup used for testing the vision-based mapping and navigation filter. A checkerboard pattern is on the face of a building as a sample target. The helicopter UAV hovers in front of the target to map it and then switches over to vision-aided inertial navigation.

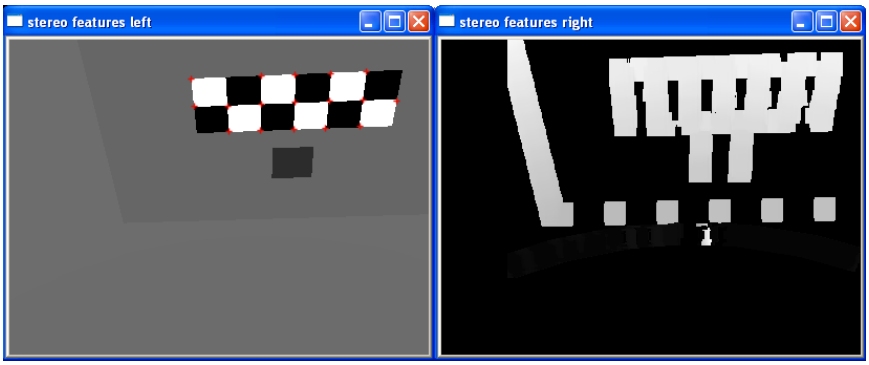

Figure 36: A sample output from the stereo image processing taken during a simulation run. The left window shows the locations of the feature points found from the corner detector, and the right window shows a grayscale representation of the computed disparity map.

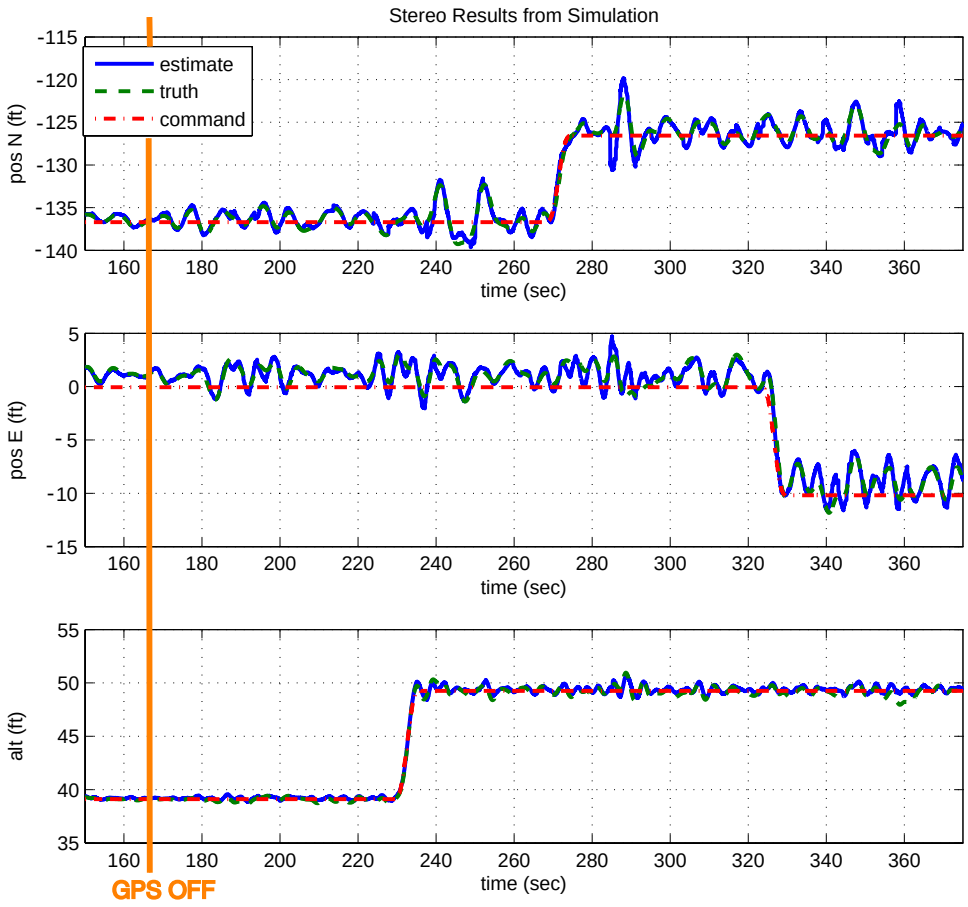

Figure 37: The estimated, actual, and commanded positions for the aircraft during a loss-of-GPS scenario with stereo vision expressed in North, East, and altitude coordinates. At $t = 166$ sec, the GPS is ignored from the navigation solution so that only the IMU and stereo vision sensor are being used for determining the pose of the helicopter for closed-loop control

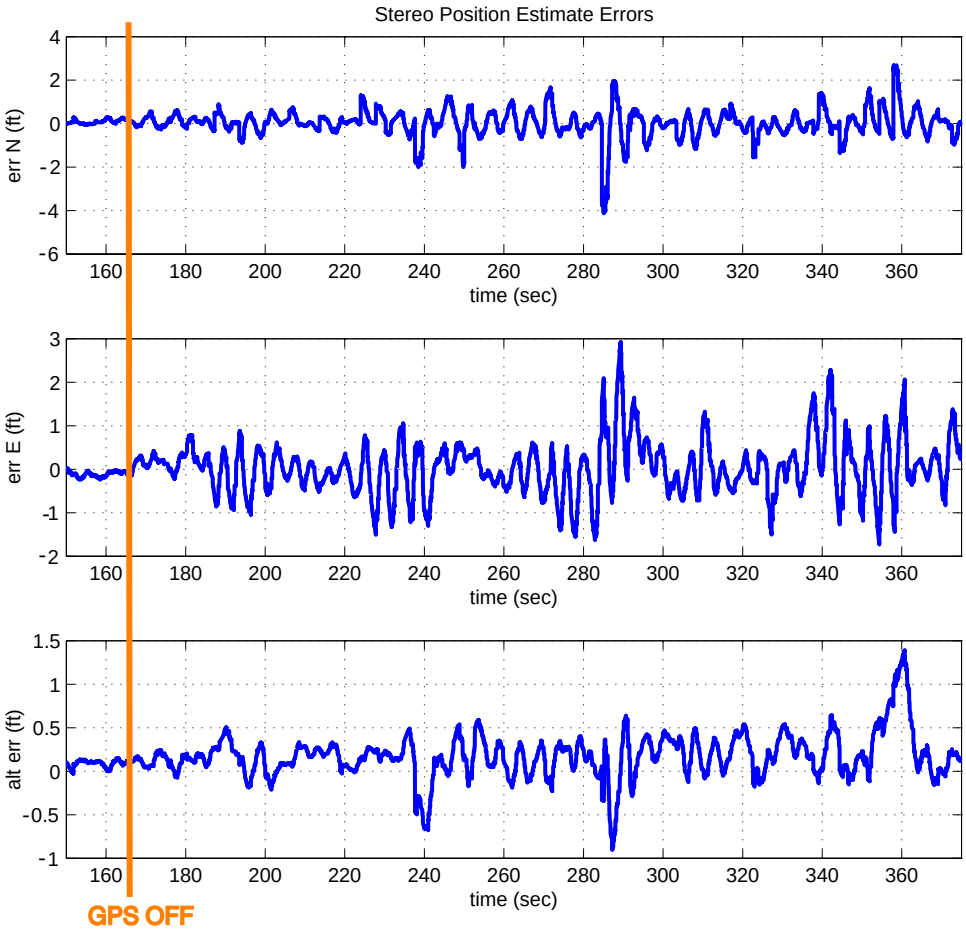

Figure 38: The error between the estimated and actual position of the helicopter during the simulation run using stereo vision.

5.5 Discussion of Method

This section provides an overall discussion of some of the details involved in the implementation of the proposed vision-based navigation and mapping algorithms.

5.5.1 Tuning the Statistical Point Correspondence

As the covariance matrix $\mathbf{P}$ for the mapping estimator decreases, the inverse term in the Z-test expression increases thereby degrading the Z-test score for points as the estimates for those landmarks improves. In order to prevent improper matches between the landmark points and their feature point measurements, the R matrix can be used to enforce a minimum value for the inverse term of the expression. The maximum Z value used as a threshold on the Z-test values to reject matches is also a term that needs adjustment. If in the environment of operation, it is found that too few correspondences are being made, then the Z_{max} can be increased, whereas too many improper matches means that the Z_{max} may need to be lowered.

5.5.2 Implementing the Estimators

Time delay compensation proved to be an important factor when using the image-based estimators. Image acquisition and processing can be time consuming tasks, and therefore may result in significant time delays between when the image was captured by the vision sensor and when the image processing results are available to the state estimator. In order to compensate for time delay, the state estimates for each of the filters were buffered so that the time delayed state estimate could be used when performing the measurement update. The correction factor to the state estimate was then applied to all other estimates in the buffer. It was also found with some extra testing that by adding in an altitude sensor, such as from a barometric altimeter or sonar, that performance of the vision-aided inertial navigation estimator was improved for the tests run since this essentially aids in the range estimation for landmarks located on the ground.

It is also worth discussing the limitations and different modes of failure for this framework. Point mismatches in the correspondence could possibly occur. However if enough landmark points are available in the surrounding environment, then the errors from sporadic mismatches can be can be compensated for if enough correct correspondences are made. It was observed from flight testing with uniform patterns that sometimes an offset in the correspondence could occur so that the vehicle might mistake the left side of a target with the right side for example. This was more prone to occurring with simpler patterns, and would result in a bias of the estimates so that the vehicle would settle into a stationary hover with a position offset from the commanded location. Another possible failure mode that could occur is that during a GPS-outage, the camera might lose sight of any landmarks that it has already visited. This might happen because the vehicle is hit suddenly by a substantial force such as a strong gust that causes a helicopter to pitch or roll hard in order to maintain position. When this occurs, no new measurement updates are available, and the algorithm boils down to an inertial-only navigation system, the performance of which depends heavily on the quality of the inertial sensors. However, if the target returns to the field of view quickly enough, then the navigation solution would return to nominal performance. This behavior was observed on occasion during the flight tests that were performed.

5.5.3 Camera Calibration

It was found that calibrating monocular vision cameras using the OpenCV camera calibration routines worked well. The results from run to run typically had a slight variation, but with a diverse sample set of around fifty images, relatively consistent results could be obtained.

However, camera calibration using the OpenCV routines for stereo vision systems proved to be more challenging. It was found that the best results were obtained by obtaining the intrinsic parameters for the cameras individually using the monocular

camera calibration routines, and then solving for the extrinsic parameters using the stereo calibration routines with the intrinsic parameters fixed constant as opposed to solving for all simultaneously with the stereo calibration function.

5.5.4 Image Processing

It was found with the current Harris corner detector that the image processing was sensitive to ambient lighting conditions. The threshold for the Harris corner metric needed to be adjusted for the time of day or to account for the weather being either sunny or overcast in order to get the desired amount of detected corners. This threshold value also affected the stability of the detected features since some would be detected consistently from image-to-image but others would be sporadic or flicker back and forth. Different sources of literature also suggested pre-processing the images by applying a Gaussian blur before performing corner detection. However, for the flight tests performed in this work, it was found that blurring the images did not provide any significant advantages. Furthermore, selecting the window size over which the Harris corner metric was computed can significantly affect the computation time for the corner detector. Using a 3x3 window worked well for the simulation and hardware tests performed.

5.5.5 Computational Complexity

The current implementation of the algorithm grows with $O((mn)^2)$ in the worst case because of the sorting that occurs in the correspondence of measurements to landmarks where m is the number of landmarks in the database being mapped and n is the number of detected feature points. There is also an $O(n^2)$ sorting that is performed during the image processing to select the best features within each grid of the image. An order of complexity analysis is useful for understanding the overall growth of the algorithm, but in practice it is not representative of the actual computations that occur. The image processing can be limited by adjusting the size of each grid

element in the image. Furthermore, the computations performed by the correspondence can be reduced by reducing the number of feature points output from the image processor. However, the number of landmark points being estimated also feeds into the correspondence, but landmark points that are not expected to be in the cameras field of view to further help reduce the computational load significantly.

CHAPTER VI

CONCLUSION

In conclusion, real-time algorithms using vision sensors to track landmark points for the purposes of navigation and mapping have been presented. A Harris corner detector was used to detect feature points, and an EKF framework was used for estimation. Points were corresponded from frame to frame using a statistical Z-test, and measurement updates were performed using sequential measurement updates to handle a variable-sized measurement array due to points dropping in and out of view.

For vision-based mapping, it was assumed that the vehicle's pose was known so that multiple observations of the same landmark point could be used to estimate the point's location in 3D inertial space. Simulation results for the algorithm were provided and the algorithm was shown to be capable of running in real-time with flight hardware available onboard a UAV. Initial estimates for the locations of landmarks in the filter could be obtained by assuming them to lie on the ground plane, but a more generic method that triangulates the points using a DLT algorithm was also presented as an alternative to allow for the initialization of points that don't fit this assumption well. This initialization method uses only knowledge of the relative change in orientation between captured images and a nearest neighbor algorithm combined with a template match to correspond points from frame to frame.

The presented vision-aided inertial navigation algorithm allows for vehicle pose to be determined when the locations of landmark points in the surrounding environment are already known. This algorithm was also capable of being run in real-time on flight hardware and was validated through application to the autonomous indoor flight of a small ducted fan UAV using only a downwards looking camera and an IMU as

sensors. The mapping and navigation filters were then integrated into a framework to assist with flight in GPS-denied environments where an aircraft can fly around with GPS while mapping features, and should GPS be lost or degraded, it can use mapped out landmarks to localize the vehicle. This was demonstrated in a loss-of-GPS scenario in real-time with a flying rotorcraft UAV. Simulation results also illustrated the extension of the methodology to a vehicle equipped with stereo vision.

6.1 Contributions of Thesis

The primary contributions of this thesis are summarized as follows:

6.1.1 Monocular Vision-Based Mapping

A vision-based mapping algorithm for estimating the 3D inertial locations of landmark points in the surrounding environment was presented. This algorithm applies to times when an aircraft knows its position and orientation in inertial space such as when GPS is available, and a standard GPS-aided INS can provide the required pose information. An EKF was used to estimate the landmark locations, and sensor measurements were provided by a monocular camera that provides 2D feature point measurements in the image plane as provided by a Harris corner detector. The correspondence of points from frame-to-frame was performed using a statistical Z-test score that tied in with the EKF. Results of the method were presented using data recorded from flight of a rotorcraft UAV. Mapped out points can potentially be used for obstacle detection and avoidance, but the primary intention was to find points that can be used as landmarks for vision-aided inertial navigation.

6.1.2 Initialization of Points in the Mapping Filter

The initialization of points in the vision-based mapping algorithm is difficult to perform with a monocular vision system since 2D measurements in the image plane do not provide any information about the relative range to a landmark point. Therefore,

a method to compute an initial guess for the mapping filter using multiple observations was presented. The method used the DLT algorithm to solve for the optimal intersection of multiple observations of a single landmark. Points could be corresponded from frame to frame using knowledge of the vehicle's change in orientation in conjunction with a nearest neighbor algorithm and a template match score. This algorithm was demonstrated using recorded images and sensor data obtained from a flight test with a rotorcraft UAV.

6.1.3 Vision-Aided Inertial Navigation

A vision-aided inertial navigation estimation filter for flight. This filter assumed the locations of landmark points in 3D space to be known, and then allowed the pose of the aircraft to be estimated using only an IMU along with observations of the landmark points from a monocular vision sensor. The vision-aided inertial navigation algorithm used the same framework as the mapping estimator, but with different states being estimated. Just as with the mapping estimator, an EKF was used along with a Harris corner detector to provide vision measurements, and points were corresponded using a statistical Z-test. One key point about the vision-aided inertial navigation scheme presented here is that if a vehicle can maintain a bounded estimate of its pose for an indefinite amount of time if the same landmarks are kept in view. This is in contrast to methods such as optical flow based ones which can accumulate errors without bound over time. A small ducted fan aircraft was flown autonomously over a known target in an indoor environment using the vision-aided inertial navigation algorithm.

6.1.4 Framework for Flight in GPS-Denied Environments

Finally, the mapping and navigation estimators were integrated into a framework to enable autonomous flight in GPS-denied environments. This was demonstrated through a loss-of-GPS scenario both in simulation and flight test with a rotorcraft UAV. The integrated methodology was also extended to the application of aircraft

equipped with stereo vision systems.

6.2 Future Work

This thesis has demonstrated the feasibility of a methodology for vision-based mapping and navigation in a simplified environment. This needs to be tested in more complex scenarios where the number of available landmark points may be highly variable ranging from very few to very many to investigate its performance. The framework also needs to be tested for its performance in acquiring new landmarks when GPS is not available so that the aircraft can continue to fly along for a sustained period of time and distance. Another aspect that could be investigated is the applicability of the framework to the SLAM problem where mapping and localization are performed simultaneously. This would be helpful for scenarios such as indoor flight where GPS is not available at any point in time to assist with mapping. However, the applicability of this algorithm to SLAM may be limited since the covariance matrix for the estimated states would grown in dimension very rapidly due to the cross-correlations between the states of the aircraft and the states of the mapped landmarks.

Vision-based sensors could also potentially integrate well with other sensors such as scanning laser range finders which have been used for flying in indoor environments when in close proximity to walls [2], [15]. Vision sensors could help to aid navigation in situations when walls are not in close enough proximity since if the vehicle is flying low enough, a downwards looking camera could still be used for navigation.

One possible improvement to the current algorithm could be to include a template matching score in the correspondence. Currently, the statistical Z-test for corresponding 3D landmark points in the database to 2D feature point measurements only takes into account the expected relative position between the aircraft and the landmark.

However, including a template-based metric allows the additional information available from the image sensor to possibly aid in the correspondence.

A variable Harris corner threshold that would maintain an approximately consistent number of detected features from scene to scene could also be included. This could potentially alleviate the tuning of the parameter due to changes in ambient lighting or significant changes in the type of surrounding scenery. Other feature point detectors could also be investigated to find one that might be able to assist with the correspondence or provide mores stable and robust feature detection. More complex landmarks in the surrounding environment could also be incorporated into the algorithm such as shapes that are common to the operating environment as opposed to point features alone. This might include windows of buildings or stop signs and other common markers in urban environments. Using these sorts of landmarks could allow for the use of existing image processing algorithms that capitalize *a priori* knowledge of these shapes for efficiency and robustness.

The results from the vision-based mapping algorithm can also be applied to the obstacle detection and avoidance problem. Currently, the scattering of points can be used to determine paths of flight least likely to contain obstructions based off the density of detected points. However, additional work remains to perform higher fidelity obstacle avoidance. Having mapped out a cloud of points, it would be ideal to be able to identify and construct the structure of obstacles from this information so that guidance laws can navigate the aircraft through them.

APPENDIX A

ATTITUDE ERROR STATE REPRESENTATION

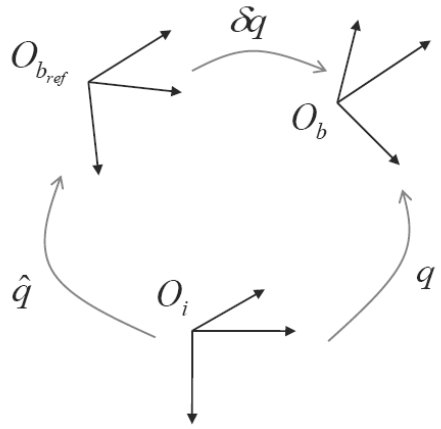

Figure 39: An attitude error quaternion can be used to express the incremental difference between a tracked reference body frame and the actual body frame for the vehicle.

The attitude of the aircraft can be represented by using a reference body frame stored as quaternions along with three error angles which represent the orientation of the aircraft with respect to the reference body frame. With each update of the state estimation filter, the reference body frame is updated to include the error angles so that the error angles are reset to zero. This means that the error angles are kept small and singularity conditions for the attitude error representation can be avoided. The attitude error can be represented as a quaternion

$$\delta\mathbf{q} = \begin{bmatrix} \cos(\delta/2) \\ n_x \sin(\delta/2) \\ n_y \sin(\delta/2) \\ n_z \sin(\delta/2) \end{bmatrix} \approx \begin{bmatrix} 1 \\ n_x(\delta/2) \\ n_y(\delta/2) \\ n_z(\delta/2) \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{1}{2}\delta\theta_1 \\ \frac{1}{2}\delta\theta_2 \\ \frac{1}{2}\delta\theta_3 \end{bmatrix} \tag{65}$$

where $\mathbf{n} = [n_x \ n_y \ n_z]^T$ is a normal direction vector and δ is the magnitude of the small rotation about the direction vector. So in order to determine the attitude of the aircraft, only the three error angles $\delta\theta = [\delta\theta_1 \ \delta\theta_2 \ \delta\theta_3]^T$ need to be estimated while the components of the quaternion for the reference body frame are tracked as auxiliary states that do not need to be explicitly estimated.

Quaternion multiplication is given by

$$\mathbf{p} \otimes \mathbf{q} = \begin{bmatrix} p_0 \\ \bar{\mathbf{p}} \end{bmatrix} \otimes \begin{bmatrix} q_0 \\ \bar{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} p_0 q_0 - \bar{\mathbf{p}} \cdot \bar{\mathbf{q}} \\ p_0 \bar{\mathbf{q}} + q_0 \bar{\mathbf{p}} + \bar{\mathbf{p}} \times \bar{\mathbf{q}} \end{bmatrix} \tag{66}$$

where p_0,q_0 are scalar and $\bar{\mathbf{p}}, \bar{\mathbf{q}} \in \mathbb{R}^3$. The kinematics for a quaternion $\mathbf{q}$ rotating with angular velocity $\omega \in \mathbb{R}^{3\times 1}$ can be written as

$$\frac{d}{dt}\mathbf{q} = \frac{1}{2}\mathbf{q} \otimes \begin{bmatrix} 0 \\ \omega \end{bmatrix} \tag{67}$$

Denoting the quaternion for the actual attitude of the vehicle as $\mathbf{q}$, the quaternion for the reference body frame as $\hat{\mathbf{q}}$, and the attitude error as $\delta\mathbf{q}$, then

$$\mathbf{q} = \hat{\mathbf{q}} \otimes \delta\mathbf{q} \tag{68}$$

differentiating gives

$$\frac{d}{dt}\mathbf{q} = \frac{d}{dt}\hat{\mathbf{q}} \otimes \delta\mathbf{q} + \hat{\mathbf{q}} \otimes \frac{d}{dt}\delta\mathbf{q} \tag{69}$$

and then defining the respective angular rates of the frames as $\omega = \hat{\omega} + \delta\omega$, the kinematic equation for quaternions from (67) can be applied to obtain

$$\frac{1}{2}\mathbf{q} \otimes \begin{bmatrix} 0 \\ \omega \end{bmatrix} = \frac{1}{2}\hat{\mathbf{q}} \otimes \begin{bmatrix} 0 \\ \hat{\omega} \end{bmatrix} \otimes \delta\mathbf{q} + \hat{\mathbf{q}} \otimes \frac{d}{dt}\delta\mathbf{q} \tag{70}$$

solving for the time rate of change of $\delta\mathbf{q}$ gives

$$\frac{d}{dt}\delta\mathbf{q} = \frac{1}{2}\hat{\mathbf{q}}^{-1} \otimes \mathbf{q} \otimes \begin{bmatrix} 0 \\ \omega \end{bmatrix} - \frac{1}{2}\begin{bmatrix} 0 \\ \hat{\omega} \end{bmatrix} \otimes \delta\mathbf{q} \tag{71}$$

$$= \frac{1}{2}\left(\delta\mathbf{q} \otimes \begin{bmatrix} 0 \\ \omega \end{bmatrix} - \begin{bmatrix} 0 \\ \hat{\omega} \end{bmatrix} \otimes \delta\mathbf{q} \right) \tag{72}$$

Using the fact that $\omega = \hat{\omega} + \delta\omega$, direct substitution gives

$$\frac{d}{dt}\delta\mathbf{q} = \frac{1}{2}\left\{\delta\mathbf{q} \otimes \left(\begin{bmatrix}0\\ \hat{\omega}\end{bmatrix} + \begin{bmatrix}0\\ \delta\omega\end{bmatrix}\right) - \begin{bmatrix}0\\ \hat{\omega}\end{bmatrix} \otimes \delta\mathbf{q}\right\} \tag{73}$$

$$= \frac{1}{2}\left\{\left(\delta\mathbf{q} \otimes \begin{bmatrix}0\\ \hat{\omega}\end{bmatrix} - \begin{bmatrix}0\\ \hat{\omega}\end{bmatrix} \otimes \delta\mathbf{q}\right) + \delta\mathbf{q} \otimes \begin{bmatrix}0\\ \delta\omega\end{bmatrix}\right\} \tag{74}$$

However, by applying the quaternion multiplication from (66) and the vector components of the incremental quaternion in (65) to these terms, it can be shown that

$$\left(\delta\mathbf{q} \otimes \begin{bmatrix}0\\ \hat{\omega}\end{bmatrix} - \begin{bmatrix}0\\ \hat{\omega}\end{bmatrix} \otimes \delta\mathbf{q}\right) = \begin{bmatrix}0\\ -\hat{\omega} \times \delta\theta\end{bmatrix} \tag{75}$$

and

$$\delta\mathbf{q} \otimes \begin{bmatrix}0\\ \delta\omega\end{bmatrix} = \begin{bmatrix}-\delta\theta \cdot \delta\omega\\ \delta\omega + \delta\theta \times \delta\omega\end{bmatrix} \approx \begin{bmatrix}0\\ \delta\omega\end{bmatrix} \tag{76}$$

where the second-order effects are dropped. Substituting these back into (74), the time rate of change of the incremental error angles are found to be

$$\frac{d}{dt}\delta\theta = -\hat{\omega} \times \delta\theta + \delta\omega \tag{77}$$

Here, $\hat{\omega}$ represents the best estimates of the vehicle angular rates in the body frame as given by the gyroscopes, and $\delta\omega$ represents the gyro biases.

To incorporate this into the EKF formulation, it remains to compute the Jacobian of the vision measurements with respect to the attitude error angles. This is found by evaluating

$$\frac{\partial\mathbf{z}}{\partial(\delta\theta)} = \left(\frac{\partial\mathbf{z}}{\partial\mathbf{r}_c}\right)\mathbf{L}_{cb}\left(\frac{\partial\mathbf{r}_b}{\partial(\delta\theta)}\right) = \left(\frac{\partial\mathbf{z}}{\partial\mathbf{r}_c}\right)\mathbf{L}_{cb}\left(\frac{\partial(\mathbf{L}_{bb_{ref}}\mathbf{r}_{b_{ref}})}{\partial(\delta\theta)}\right) \tag{78}$$

The first two matrices in the product expression were found before, so it remains to find the third part. By generating the coordinate transformation matrix for the incremental quaternion using equation (36) and neglecting second order effects, it is

found that

$$\mathbf{L}_{bb_{ref}}\mathbf{r}_{b_{ref}} = \begin{bmatrix} 1 & \delta\theta_3 & -\delta\theta_2 \\ -\delta\theta_3 & 1 & \delta\theta_1 \\ \delta\theta_2 & -\delta\theta_1 & 1 \end{bmatrix} \mathbf{r}_{b_{ref}} \tag{79}$$

Expanding this and computing the partial derivatives with respect to the attitude error angles, it can be shown that

$$\frac{\partial(\mathbf{L}_{bb_{ref}}\mathbf{r}_{b_{ref}})}{\partial(\delta\theta)} = \tilde{\mathbf{r}}_{b_{ref}} \tag{80}$$

APPENDIX B

DATA NORMALIZATION FOR DLT TRIANGULATION

In the DLT algorithm for determining an initial guess for points in the mapping database with multiple corresponded observations of a landmark using a monocular camera, data normalization helps to make the elements of the $\mathbf{A}_{dlt}$ matrix have the same order of magnitude to improve accuracy. Data normalization also helps to make the algorithm invariant to choices of scale and coordinate origin. Hartley and Zisserman [21] claim that data normalization for the DLT algorithm is not an optional step and they propose a scaling and translation scheme to normalize the data matrix. Using their normalization scheme, the set of measurements for a landmark can be normalized by defining the following averages

$$\bar{x} = \frac{1}{N}\sum_{k=1}^{N} x_k \tag{81}$$

$$\bar{y} = \frac{1}{N}\sum_{k=1}^{N} y_k \tag{82}$$

$$\bar{d} = \frac{1}{N}\sum_{k=1}^{N} \sqrt{(x_k-\bar{x})^2+(y_k-\bar{y})^2} \tag{83}$$

and then constructing the following normalization matrix $\mathbf{T}$ as

$$\mathbf{T} = \begin{bmatrix} \sqrt{2}/\bar{d} & 0 & -\bar{x}/\bar{d} \\ 0 & \sqrt{2}/\bar{d} & -\bar{y}/\bar{d} \\ 0 & 0 & 1 \end{bmatrix} \tag{84}$$

the set of normalized data points are given as

$$\bar{\mathbf{z}}'_k = \mathbf{T}\bar{\mathbf{z}}_k \tag{85}$$

This makes it so that the collection of 2D points $\mathbf{z}'_k$ has a centroid at the coordinate origin (0,0) and an average distance of $\sqrt{2}$ from the origin.

Similarly, points in 3D space can be normalized by defining the averages

$$\bar{x} = \frac{1}{N}\sum_{k=1}^{N} x_k \tag{86}$$

$$\bar{y} = \frac{1}{N}\sum_{k=1}^{N} y_k \tag{87}$$

$$\bar{z} = \frac{1}{N}\sum_{k=1}^{N} z_k \tag{88}$$

$$\bar{d} = \frac{1}{N}\sum_{k=1}^{N} \sqrt{(x_k-\bar{x})^2+(y_k-\bar{y})^2+(z_k-\bar{z})^2} \tag{89}$$

and then constructing the following normalization matrix $\mathbf{U}$ as

$$\mathbf{U} = \begin{bmatrix} \sqrt{3}/\bar{d} & 0 & 0 & -\bar{x}/\bar{d} \\ 0 & \sqrt{3}/\bar{d} & 0 & -\bar{y}/\bar{d} \\ 0 & 0 & \sqrt{3}/\bar{d} & -\bar{z}/\bar{d} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{90}$$

the set of normalized locations are given as

$$\bar{\mathbf{p}}'_{i_k} = \mathbf{U}\bar{\mathbf{p}}_{i_k} \tag{91}$$

This makes it so that the collection of 3D points $\mathbf{p}'_{i_k}$ has a centroid at the origin (0,0,0) and an average distance of $\sqrt{3}$ from the origin. Note however, that in order to express the scaling and translation, the normalization matrix can also be expressed as

$$\mathbf{U} = \begin{bmatrix} \mathbf{D} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \tag{92}$$

where $\mathbf{D} \in \mathbb{R}^{3\times 3}$ is the diagonal scaling matrix and $\mathbf{t} \in \mathbb{R}^{3\times 1}$ is the translation vector.

These can be applied to the DLT algorithm by normalizing each of the measurements using the 2D $\mathbf{T}$ matrix, and normalizing the camera matrix by applying the

3D normalization matrix $\mathbf{U}$ to the position vectors

$$\begin{aligned}
\bar{\mathbf{z}} &= \mathbf{\Phi}\bar{\mathbf{p}}_{fp_i} && (93)\\
&= \mathbf{\Gamma}\mathbf{L}_{ci}\left[\mathbf{I} \mid -\mathbf{p}_i\right]\bar{\mathbf{p}}_{fp_i} && (94)\\
&= \mathbf{\Gamma}\mathbf{L}_{ci}\left[\mathbf{I} \mid -\mathbf{D}^{-1}(\mathbf{D}\mathbf{p}_i + \mathbf{t} - \mathbf{t})\right]\bar{\mathbf{p}}_{fp_i} && (95)\\
&= \mathbf{\Gamma}\mathbf{L}_{ci}\mathbf{D}^{-1}\left[\mathbf{D} \mid -(\mathbf{D}\mathbf{p}_i + \mathbf{t}) + \mathbf{t}\right]\bar{\mathbf{p}}_{fp_i} && (96)\\
&= \mathbf{\Gamma}\mathbf{L}_{ci}\mathbf{D}^{-1}\left[\mathbf{I} \mid -(\mathbf{D}\mathbf{p}_i + \mathbf{t})\right]\begin{bmatrix}\mathbf{D}\mathbf{p}_{fp_i} + \mathbf{t}\\ 1\end{bmatrix} && (97)
\end{aligned}$$

Applying the normalization

$$\begin{aligned}
\bar{\mathbf{z}}' &= \mathbf{T}\mathbf{\Gamma}\mathbf{L}_{ci}\mathbf{D}^{-1}\left[\mathbf{I} \mid -(\mathbf{D}\mathbf{p}_i + \mathbf{t})\right]\bar{\mathbf{p}}'_{fp_i} && (98)\\
&= \mathbf{P}'\bar{\mathbf{p}}'_{fp_i} && (99)
\end{aligned}$$

which means that the DLT algorithm can be applied to the equation $\bar{\mathbf{z}}' = \mathbf{P}'\bar{\mathbf{p}}'_{fp_i}$ to solve for the normalized $\bar{\mathbf{p}}'_{fp_i}$ which allows the location of the triangulated landmark to be computed as $\bar{\mathbf{p}}_{fp_i} = \mathbf{U}^{-1}\bar{\mathbf{p}}'_{fp_i}$

REFERENCES

[1] "Opencv computer vision library." `http://opencv.willowgarage.com/wiki/`.

[2] Achtelik, M., Bachrach, A., He, R., Prentice, S., and Roy, N., "Stereo vision and laser odometery for authomous helicopters in gps-denied indoor environments," in *Proceedings of SPIE - The International Society for Optical Engineering*, 2009.

[3] Ahrens, S., Levine, D., Andrews, G., and How, J. P., "Vision-based guidance and control of a hovering vehicle in unknown, gps-denied environments," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2009.

[4] Amidi, O., Kanade, T., and Fujita, K., "A visual odometer for autonomous helicopter flight," *Robotics and Autonomous Systems*, vol. 28, pp. 185–193, 1999.

[5] Barber, B., McLain, T., and Edwards, B., "Vision-based landing of fixed-wing miniature air vehicles," *AIAA Journal of Aerospace Computing, Information, and Communication*, vol. 6, no. 3, pp. 207–226, 2009.

[6] Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V., "Surf: Speeded up robust features," *Computer Vision and Image Understanding*, vol. 110, pp. 246–359, Sep 2008.

[7] Bernatz, A. and Thielecke, F., "Navigation of a low flying vtol aircraft with the help of a downwards pointing camera," in *AIAA Guidance, Navigation, and Control Conference Proceedings*, 2004.

[8] Brink, K., Hurtado, J., Soloviev, A., Rutkowski, A., and Klausutis, T., "Integrated estimation and control approach for vision aided inertial navigation," in *AIAA Infotech@Aerospace Proceedings*, 2010.

[9] Brown, D. C., "Close-range camera calibration," *Photogrammetric Engineering*, vol. 37, no. 8, pp. 855–866, 1971.

[10] Caballero, F., Merino, L., Ferruz, J., and Ollero, A., "Vision-based odometry and slam for medium and high altitude flying uavs," *Journal of Intelligent and Robotic Systems*, vol. 54, pp. 137–161, March 2008.

[11] Call, B., Beard, R., Taylor, C., and Barber, B., "Obstacle avoidance for unmanned air vehicles using image feature tracking," in *AIAA Guidance, Navigation, and Control Conference Proceedings*, 2006.

[12] Celik, K., Chung, S.-J., Clausman, M., and Somani, A. K., "Biologically inspired monocular vision based navigation and mapping in gps-denied environments," in *AIAA Infotech@Aerospace Proceedings*, 2009.

[13] Christophersen, H., Pickell, R. W., Neidhoefer, J., Koller, A. A., Kannan, S. K., and Johnson, E. N., "A compact guidance, navigation, and control system for unmanned aerial vehicles," *Journal of Aerospace Computing, Information, and Communication*, vol. 3, no. 5, pp. 187–213, 2006.

[14] Conte, G. and Doherty, P., "Vision-based unmanned aerial vehicle navigation using geo-referenced information," *EURASIP Journal of Advances in Signal Processing*, vol. 2009, no. 387308, p. 18 pages, 2009.

[15] D. Michael Sobers, J., Yamaura, S., and Johnson, E. N., "Laser-aided inertial navigation for self-contained autonomous indoor flight," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2010.

[16] Etkin, B., *Dynamics of Atmospheric Flight*. John Wiley & Sons, Inc., 1972.

[17] Frietsch, N., Meister, O., Schlaile, C., Seibold, J., and Trommer, G. F., "Vision based hovering and landing system for a vtol-mav with geo-localization capabilities," in *AIAA Guidance, Navigation, and Control Conference Proceedings*, 2008.

[18] Gariepy, R. and Waslander, S. L., "Uav motion estimation using low quality image features," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2010.

[19] Griffiths, S., Saunders, J., Curtis, A., Barber, B., McLain, T., and Beard, R., "Maximizing miniature aerial vehicles," *IEEE Robotics & Automation Magazine*, vol. 13, pp. 34–43, Sep 2006.

[20] Harris, C. and Stephens, M., "A combined corner and edge detector," in *Alvey Vision Conference Proceedings*, 1998.

[21] Hartley, R. and Zisserman, A., *Multiple View Geometry in Computer Vision*. Cambridge University Press, second ed., 2003.

[22] Herisse, B., Oustrieres, S., Hamel, T., Mahoney, R., and Russotto, F. X., "A general optical flow based terrain-following strategy for a vtol uav using multiple views," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2010.

[23] Hrabar, S., Sukhatme, G. S., Corke, P., Usher, K., and Roberts, J., "Combined optic-flow and stereo-based navigation of urban canyons for a uav," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

[24] Ivey, G. F. and Johnson, E. N., "Investigation of methods for simultaneous localization and mapping using vision sensors," in *AIAA Guidance, Navigation, and Control Conference Proceedings*, 2006.

[25] Jensfelt, P., Kragic, D., Folkesson, J., and Björkman, M., "A framework for vision based bearing only 3d slam," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006.

[26] Johnson, E. N. and Kannan, S. K., "Adaptive trajectory control for autonomous helicopters," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 28, pp. 524–538, May-June 2005.

[27] Johnson, E. N. and Schrage, D. P., "The georgia tech unmanned aerial research vehicle: Gtmax," in *AIAA Guidance, Navigation, and Control Conference Proceedings*, 2003.

[28] Kanade, T., Amidi, O., and Ke, Q., "Real-time and 3d vision for autonomous small and micro air vehicles," in *IEEE Conference on Decision and Control Proceedings*, 2004.

[29] Kannan, S. K., Koller, A. A., and Johnson, E. N., "Simulation and development environment for multiple heterogeneous uavs," in *Proceedings of the AIAA Modeling and Simulation Technologies Conference*, 2004.

[30] Kehoe, J. J., Watkins, A. S., Causey, R. S., and Lind, R., "State estimation using optical flow from parallax-weighted feature tracking," in *AIAA Guidance, Navigation, and Control Conference Proceedings*, 2006.

[31] Kelly, J., Saripalli, S., and Sukhatme, G. S., "Combined visual and inertial navigation for an unmanned aerial vehicle," *Springer Tracts in Advanced Robotics*.

[32] Kim, J. and Sukkarieh, S., "Autonomous airborne navigation in unknown terrain environments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 3, 2004.

[33] Koch, A., Wittich, H., and Thielecke, F., "A vision-based navigation algorithm for a vtol-uav," in *AIAA Guidance, Navigation, and Control Conference Proceedings*, 2006.

[34] Kriegman, D. J., Triendl, E., and Binford, T. O., "Stereo vision and navigation in buildings for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 6, pp. 792–803, 1989.

[35] Langelaan, J. W., "State estimation for autonomous flight in cluttered environments," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 30, pp. 1414–1426, Sep–Oct 2007.

[36] Lefferts, E., Markley, F., and Shuster, M., "Kalman filtering for spacecraft attitude estimation," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 5, pp. 417–429, Sep–Oct 1982.

[37] Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, Nov 2004.

[38] Madison, R., Andrews, G., DeBitetto, P., Rasmussen, S., and Bottkol, M., "Vision-aided navigation for small uavs in gps-challenged environments," in *AIAA Infotech@Aerospace Proceedings*, 2007.

[39] Matthies, L. and Shafer, S. A., "Error modeling in stereo navigation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 3, pp. 239–248, 1987.

[40] Meingast, M., Geyer, C., and Sastry, S., "Vision based terrain recovery for landing unmanned aerial vehicles," in *Proceedings of the IEEE Conference on Decision and Control*, 2004.

[41] Merz, T., Duranti, S., and Conte, G., "Autonomous landing of an unmanned helicopter based on vision and inertial sensing," in *Experimental Robotics IX* (Ang, M. and Khatib, O., eds.), vol. 21 of *Springer Tracts in Advanced Robotics*, pp. 343–352, Springer Berlin / Heidelberg, 2006.

[42] Mkrtchyan, A. A., Schultz, R. R., and Semke, W. H., "Vision-based autopilot implementation using a quadrotor helicopter," in *AIAA Infotech@Aerospace Proceedings*, 2009.

[43] Ortiz, A. E. and Neogi, N. N., "Object detection and avoidance using optical techniques in uninhabited aerial vehicles," in *AIAA Guidance, Navigation, and Control Conference Proceedings*, 2007.

[44] Piniés, P., Lupton, T., Sukkarieh, S., and Tardós, J. D., "Inertial aiding of inverse depth slam using a monocular camera," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2007.

[45] Pollini, L., Greco, F., Mati, R., Innocenti, M., and Tortelli, A., "Stereo vision obstacle detection based on scale invariant feature transform algorithm," in *AIAA Guidance, Navigation, and Control Conference Proceedings*, 2007.

[46] Prazenica, R. J., Watkins, A., Kurdila, A. J., Ke, Q. F., and Kanade, T., "Vision-based kalman filtering for aircraft state estimation and structure from motion," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2005.

[47] Proctor, A. A. and Johnson, E. N., "Vision-only aircraft flight control methods and test results," in *AIAA Guidance, Navigation, and Control Conference Proceedings*, 2004.

[48] Proctor, A. A. and Johnson, E. N., "Vision-only approach and landing," in *AIAA Guidance, Navigation, and Control Conference Proceedings*, 2005.

[49] Saripalli, S., Montgomery, J. F., and Sukhatme, G. S., "Vision-based autonomous landing of an unmanned aerial vehicle," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2002.

[50] Schulz, H., Buschmann, M., Krüger, L., Winkler, S., and Vörsmann, P., "Towards vision-based autonomous landing for small uavs - first experimental results of the vision system," *AIAA Journal of Aerospace Computing, Information, and Communication*, vol. 4, no. 5, pp. 785–797, 2007.

[51] Shah, S. I. A., Kannan, S., and Johnson, E. N., "Motion estimation for obstacle detection and avoidance using a single camera for uavs/robots," in *AIAA Guidance, Navigation, and Control Conference Proceedings*, 2010.

[52] Shah, S. I. A. and Johnson, E. N., "3d obstacle detection using a single camera," in *AIAA Guidance, Navigation, and Control Conference Proceedings*, 2009.

[53] Sharp, C. S., Shakernia, O., and Sastry, S. S., "A vision system for landing an unmanned aerial vehicle," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2001.

[54] Sinopoli, B., Micheli, M., Donato, G., and Koo, T. J., "Vision based navigation for an unmanned aerial vehicle," in *Proceedings of the IEEE International Conference on Robotics and Automation*.

[55] Soundararaj, S. P., Sujeeth, A. K., and Saxena, A., "Autonomous indoor helicopter flight using a single onboard camera," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[56] Steder, B., Grisetti, G., Stachniss, C., and Burgard, W., "Visual slam for flying vehicles," *IEEE Transactions on Robotics and Automation*, vol. 24, no. 5, pp. 1088–1093, 2008.

[57] Stevens, B. L. and Lewis, F. L., *Aircraft Control and Simulation*. John Wiley & Sons, Inc., second ed., 2003.

[58] Tippetts, B. J., Lee, D., Fowers, S. G., and Archibald, J. K., "Real-time vision sensor for an autonomous hovering micro unmanned aerial vehicle," *AIAA Journal of Aerospace Computing, Information, and Communication*, vol. 6, no. 10, pp. 570–584, 2009.

[59] Toupet, O., Paduano, J., Panish, R., Sedwick, R., and Frazzoli, E., "Augmenting state estimates with multiple camera visual measurements," in *AIAA Infotech@Aerospace Proceedings*, 2007.

[60] Tournier, G. P., Valenti, M., How, J. P., and Feron, E., "Estimation and control of a quadrotor vehicle using monocular vision and moiré patterns," in *AIAA Guidance, Navigation, and Control Conference Proceedings*, 2006.

[61] Wagter, C. D. and Mulder, J., "Towards vision-based uav situation awareness," in *AIAA Guidance, Navigation, and Control Conference Proceedings*, 2005.

[62] Watanabe, Y., Johnson, E. N., and Calise, A. J., "Vision-based approach to obstacle avoidance," in *AIAA Guidance, Navigation, and Control Conference Proceedings*, 2005.

[63] Watanabe, Y., Johnson, E. N., and Calise, A. J., "Stoachastic guidance design for uav vision-based control applications," in *AIAA Guidance, Navigation, and Control Conference Proceedings*, 2008.

[64] Watkins, A. S., Kehoe, J. J., and Lind, R., "Slam for flight through urban environments using dimensionality reduction," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2006.

[65] Webb, T. P., Prazenica, R. J., Kurdila, A. J., and Lind, R., "Vision-based state estimation for autonomous micro air vehicles," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 30, pp. 816–826, May–June 2007.

[66] Wu, A. D., Johnson, E. N., and Proctor, A. A., "Vision-aided inertial navigation for flight control," *AIAA Journal of Aerospace Computing, Information, and Communication*, vol. 2, pp. 348–360, Sep 2005.